

888888888888		AAAAA		CCCCCCCCCCCC	KKK		KKK	UUU		UUU	PPPPPPPPPPPP	
888888888888		AAAAA		CCCCCCCCCCCC	KKK		KKK	UUU		UUU	PPPPPPPPPPPP	
888888888888		AAAAA		CCCCCCCCCCCC	KKK		KKK	UUU		UUU	PPPPPPPPPPPP	
888	888	AAA	AAA	CCC	KKK		KKK	UUU		UUU	PPP	PPP
888	888	AAA	AAA	CCC	KKK		KKK	UUU		UUU	PPP	PPP
888	888	AAA	AAA	CCC	KKK		KKK	UUU		UUU	PPP	PPP
888	888	AAA	AAA	CCC	KKK	KKK	KKK	UUU		UUU	PPP	PPP
888	888	AAA	AAA	CCC	KKK	KKK	KKK	UUU		UUU	PPP	PPP
888	888	AAA	AAA	CCC	KKK	KKK	KKK	UUU		UUU	PPP	PPP
888888888888		AAA	AAA	CCC	KKKKKKKKKK			UUU		UUU	PPPPPPPPPPPP	
888888888888		AAA	AAA	CCC	KKKKKKKKKK			UUU		UUU	PPPPPPPPPPPP	
888888888888		AAA	AAA	CCC	KKKKKKKKKK			UUU		UUU	PPPPPPPPPPPP	
888	888	AAAAAAAAAAAA		CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAAAAAAAAAAA		CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAAAAAAAAAAA		CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAA	AAA	CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAA	AAA	CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAA	AAA	CCC	KKK	KKK		UUU		UUU	PPP	
888	888	AAA	AAA	CCC	KKK	KKK		UUU		UUU	PPP	
888888888888		AAA	AAA	CCCCCCCCCCCC	KKK	KKK		UUUUUUUUUUUUUUUU		UUU	PPP	
888888888888		AAA	AAA	CCCCCCCCCCCC	KKK	KKK		UUUUUUUUUUUUUUUU		UUU	PPP	
888888888888		AAA	AAA	CCCCCCCCCCCC	KKK	KKK		UUUUUUUUUUUUUUUU		UUU	PPP	

```

LL                      IIIII
LL                      IIIII
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LLLLLLLLLLLL           IIIII
LLLLLLLLLLLL           IIIII

SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```

```

1 0001 0 MODULE RESTORE (XTITLE 'Restore a save set'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Backup/Restore
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains the routines that restore a save set.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1 VAX/VMS user mode.
40 0040 1 --
41 0041 1
42 0042 1 AUTHOR: M. Jack, CREATION DATE: 03-Sep-1980
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-025 LY0513 Larry Yetto 23-JUL-1984 15:36
47 0047 1 Fix incremental restore's so that they do not look at the default
48 0048 1 directory and hence place files in the incorrect location.
49 0049 1
50 0050 1 V03-024 LY0509 Larry Yetto 19-JUL-1984 08:39
51 0051 1 Add support for HSC generated nonsequential physical save sets
52 0052 1
53 0053 1 V03-023 LMP0272 L. Mark Pilant, 6-Jul-1984 8:53
54 0054 1 Modify BACKUP to always use a full FIB.
55 0055 1
56 0056 1 V03-022 LY0481 Larry Yetto 27-APR-1984 08:34
57 0057 1 FT1 QAR # 2088 - If the saveset being read is

```

58	0058	1	encrypted then report an error if /ENCRYPT was not
59	0059	1	specified.
60	0060	1	FT1 QAR # 1847 - Correct directory protection on newly
61	0061	1	created directly
62	0062	1	FT1 QAR #1880 - Do not attempt to mod ACL's on verify
63	0063	1	pass
64	0064	1	
65	0065	1	V03-021 LY0478 Larry Yetto 18-APR-1984 15:49
66	0066	1	Correct length field in the attribute descriptor used
67	0067	1	in the QIO/IOS_MODIFY to add ACL's
68	0068	1	
69	0069	1	V03-020 LY0472 Larry Yetto 9-APR-1984 08:23
70	0070	1	Modify the CREATE_DIRECTORY routine so that it leaves the directory
71	0071	1	file open when it is done. This is necessary so that any ACL's
72	0072	1	on the old file can be propagated to the new one.
73	0073	1	
74	0074	1	V03-019 LY0456 Larry Yetto 1-FEB-1984 10:16
75	0075	1	Make restore operation restartable
76	0076	1	
77	0077	1	V03-018 ACG0380 Andrew C. Goldstein, 9-Dec-1983 19:27
78	0078	1	Tolerate zero file ID in file attribute record
79	0079	1	
80	0080	1	V03-017 ACG0373 Andrew C. Goldstein, 12-Nov-1983 19:04
81	0081	1	Bug fixes to /INCREMENTAL handling to deal with
82	0082	1	dangling directory entries. Also swallow \$\$\$_DIRNOTEMPTY
83	0083	1	errors in the incremental delete pass.
84	0084	1	
85	0085	1	V03-016 ACG0343 Andrew C. Goldstein, 19-Jul-1983 10:26
86	0086	1	Use NORECORD option to set backup dates; make /INCREMENTAL
87	0087	1	imply /OWNER=ORIGINAL
88	0088	1	
89	0089	1	V03-015 LMP0127 L. Mark Pilant, 6-Jul-1983 9:55
90	0090	1	Correct a bug that caused the ACL not to be written because
91	0091	1	of using the wrong I/O function codes.
92	0092	1	
93	0093	1	V03-014 ACG0340 Andrew C. Goldstein, 22-Jun-1983 18:22
94	0094	1	Copy last track of MSCP disks in /PHYSICAL operations
95	0095	1	
96	0096	1	V03-013 JEP0003 J. Eric Pollack, 23-Apr-1983 10:53
97	0097	1	Add support for encrypted savesets.
98	0098	1	
99	0099	1	V03-012 ACG0332 Andrew C. Goldstein, 20-Apr-1983 16:30
100	0100	1	Add RMS journal flags
101	0101	1	
102	0102	1	V03-011 ACG0313 Andrew C. Goldstein, 12-Feb-1983 16:28
103	0103	1	Add routine subtitles
104	0104	1	
105	0105	1	V03-010 MLJ0104 Martin L. Jack, 24-Jan-1983 23:45
106	0106	1	Clear the backup date of restored files except under /IMAGE,
107	0107	1	/INCREMENTAL, or /RECORD.
108	0108	1	
109	0109	1	V03-009 MLJ50598 Martin L. Jack, 13-Dec-1982 19:55
110	0110	1	Repair selected volume copy with verify.
111	0111	1	
112	0112	1	V03-008 LMP0044 L. Mark Pilant, 21-Oct-1982 15:45
113	0113	1	Add support for saving and restoring ACL's.
114	0114	1	

RESTORE
V04-000

Restore a save set

H 4
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 3
(1)

115	0115	1	V03-007	MLJ48500	Martin L. Jack, 17-Aug-1982 14:20
116	0116	1			Add parentheses to prevent INCREMENTAL_SCAN from being called
117	0117	1			at the wrong time.
118	0118	1			
119	0119	1	V03-006	LMP0032	L. Mark Pilant, 22-Jun-1982 10:30
120	0120	1			Add support for wildcard save set names on a LIST or RESTORE
121	0121	1			operation.
122	0122	1			
123	0123	1	V03-005	MLJ0089	Martin L. Jack, 22-Apr-1982 23:27
124	0124	1			Repair /CONFIRM for copy operation and extra directories.
125	0125	1			
126	0126	1	V03-004	MLJ43240	Martin L. Jack, 6-Apr-1982 2:46
127	0127	1			Avoid spurious EOFMISMATCH messages on a restore verify pass.
128	0128	1			
129	0129	1	V03-003	ACG0277	Andrew C. Goldstein, 30-Mar-1982 16:55
130	0130	1			Add BACKUPS_SSCHANGE to list of continuable errors
131	0131	1			
132	0132	1	V03-002	MLJ0084	Martin L. Jack, 28-Mar-1982 1:08
133	0133	1			Update the backup date of a directory file that was scanned
134	0134	1			during a /INCREMENTAL restore to record that it represents the
135	0135	1			most recent version. Readjust the order of operations so that
136	0136	1			/OWNER=PARENT with an implicit directory creation works.
137	0137	1			
138	0138	1	V03-001	MLJ0082	Martin L. Jack, 16-Mar-1982 13:27
139	0139	1			Correct /OWNER=PARENT problem introduced by V02-018. Extend
140	0140	1			NOFILES message to contain save set name string.
141	0141	1			
142	0142	1	V02-020	MLJ0075	Martin L. Jack, 28-Jan-1982 20:35
143	0143	1			Use FIBSV NORECORD. Implement version limit handling via
144	0144	1			DIR_VERLIM and VERLIMIT attributes.
145	0145	1			
146	0146	1	V02-019	MLJ0072	Martin L. Jack, 19-Jan-1982 3:42
147	0147	1			Extend if existing file opened under /OVERLAY is too small.
148	0148	1			
149	0149	1	V02-018	MLJ42936	Martin L. Jack, 5-Jan-1982 19:40
150	0150	1			Create files initially with the correct owner UIC to avoid
151	0151	1			temporarily consuming (and possibly exceeding) the disk quota
152	0152	1			of the process executing a restore.
153	0153	1			
154	0154	1	V02-017	MLJ0062	Martin L. Jack, 2-Dec-1981 13:12
155	0155	1			Implement /INCREMENTAL. Avoid spurious VBNMISSING error on
156	0156	1			tape verify pass.
157	0157	1			
158	0158	1	V02-016	MLJ0054	Martin L. Jack, 15-Oct-1981 19:54
159	0159	1			Attempt non-contiguous create if contiguous create fails.
160	0160	1			Implement /VOLUME. Implement /IGNORE=INTERLOCK. Correct
161	0161	1			V02-014 to include EFBLK.
162	0162	1			
163	0163	1	V02-015	MLJ0053	Martin L. Jack, 7-Oct-1981 19:27
164	0164	1			Rework algorithm in CHECK_EOF_BLOCK.
165	0165	1			
166	0166	1	V02-014	MLJ0048	Martin L. Jack, 28-Sep-1981 13:14
167	0167	1			Read attributes before IOS_MODIFY since standalone ACP
168	0168	1			no longer protects HIBLK.
169	0169	1			
170	0170	1	V02-013	MLJ0040	Martin L. Jack, 3-Sep-1981 19:37
171	0171	1			Correct message following IOS_CREATE.

RESTORE
V04-000

Restore a save set

I 4
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 4
(1)

172	0172	1	
173	0173	1	
174	0174	1	V02-012 MLJ0036 Martin L. Jack, 29-Aug-1981 16:09
175	0175	1	Extensive rewriting to complete implementation.
176	0176	1	
177	0177	1	V02-011 MLJ0025 Martin L. Jack, 8-May-1981 14:00
178	0178	1	Reorganize qualifier database. Correct problem with explicit
179	0179	1	directory creation introduced by V02-009.
180	0180	1	
181	0181	1	V02-010 MLJ0024 Martin L. Jack, 27-Apr-1981 20:25
182	0182	1	Remove dynamic LOCAL_SAVE area, since it can now be local.
183	0183	1	
184	0184	1	V02-009 MLJ0022 Martin L. Jack, 21-Apr-1981 13:54
185	0185	1	Convert directory wildcarding to use RMS.
186	0186	1	
187	0187	1	V02-008 MLJ0021 Martin L. Jack, 20-Apr-1981 22:04
188	0188	1	Implement /TRUNCATE qualifier.
189	0189	1	
190	0190	1	V02-007 MLJ0019 Martin L. Jack, 7-Apr-1981 22:23
191	0191	1	Correct tests for continuation tape verify pass.
192	0192	1	
193	0193	1	V02-006 MLJ0018 Martin L. Jack, 7-Apr-1981 21:09
194	0194	1	Correct signal parameters.
195	0195	1	
196	0196	1	V02-005 MLJ0014 Martin L. Jack, 6-Apr-1981 23:27
197	0197	1	Correct and clarify VBN sequence-error messages and
198	0198	1	avoid spurious EOFMISMATCH messages on tape verify pass.
199	0199	1	
200	0200	1	V02-004 MLJ0012 Martin L. Jack, 27-Mar-1981 16:14
201	0201	1	Produce both CREATED and COMPARED messages for copy with verify
202	0202	1	
203	0203	1	V02-003 MLJ0010 Martin L. Jack, 25-Mar-1981 17:29
204	0204	1	Reorganize global storage. Changes for standalone operation.
205	0205	1	Implement image restore. Implement listing concurrent with
206	0206	1	another operation.
207	0207	1	
208	0208	1	V02-002 MLJ0003 Martin L. Jack, 17-Feb-1981 18:36
209	0209	1	Correct handling of DNF in compare operation
210	0210	1	
211	0211	1	V02-001 MLJ0002 Martin L. Jack, 14-Feb-1981 12:23
212	0212	1	Correct file selection in restore operation verify pass
213	0213	1	★★

```
215 0214 1 REQUIRE 'SRCS:COMMON';
216 1320 1 LIBRARY 'SYS$LIBRARY:LIB';
217 1321 1 REQUIRE 'LIB$BACKDEF';
218 1771 1
219 1772 1
220 1773 1 FORWARD ROUTINE
221 1774 1 INCREMENTAL_DELETE:
222 1775 1 NOVALUE, ! Conditionally delete file
223 1776 1 INCREMENTAL_SCAN:
224 1777 1 NOVALUE, ! Execute wildcard access on directory
225 1778 1 ASSIGN_VERIFY_CHANNEL:
226 1779 1 NOVALUE, ! Assign channel to input device for verify
227 1780 1 COMPARE_BLOCKS: NOVALUE, ! Compare two files
228 1781 1 GET_ACP_ATTRIBUTES:
229 1782 1 NOVALUE, ! Convert OUTPUT_ATTBUF for write attributes
230 1783 1 GET_PARENT_UIC, ! Get owner UIC of parent directory
231 1784 1 CLOSE_RESTORE: NOVALUE, ! Close output file
232 1785 1 FIN_RESTORE: NOVALUE, ! Finish restore operation
233 1786 1 SELECT_FILE, ! Evaluate output selection criteria
234 1787 1 CHECK_EOF_BLOCK:NOVALUE, ! Check EOF position in a compare
235 1788 1 CREATE_DIRECTORY:
236 1789 1 NOVALUE, ! Create a directory
237 1790 1 IMPLICIT_DIRECTORY, ! Implicitly create a directory (DNF)
238 1791 1 INITIALIZE AND SELECT, ! Initialize FAB and select file
239 1792 1 SELECT_INCREMENTAL, ! File selection for /INCREMENTAL mode
240 1793 1 INCREMENTAL_DIRECTORY:
241 1794 1 NOVALUE, ! Directory processing for /INCREMENTAL
242 1795 1 CREATE_OR_OPEN_FILE:
243 1796 1 NOVALUE, ! Open output file
244 1797 1 RESTORE_ERROR: NOVALUE, ! Handle asynchronous write errors
245 1798 1 RESTORE_ONE_RECORD:
246 1799 1 NOVALUE, ! Process one save set buffer
247 1800 1 RESTORE_VERIFY_REEL:
248 1801 1 NOVALUE, ! Driver for restore verify
249 1802 1 INIT_RESTORE: NOVALUE, ! Initialize restore module
250 1803 1 RESTORE_HANDLER, ! Handler for RESTORE, LIST, ANALYZE
251 1804 1 RESTORE: NOVALUE; ! Driver for restore
252 1805 1
253 1806 1
254 1807 1 EXTERNAL ROUTINE
255 1808 1 LIB$CVT_DTB: ADDRESSING_MODE(GENERAL),
256 1809 1 ! Convert text to binary
257 1810 1 LIB$GET_COMMAND:ADDRESSING_MODE(GENERAL),
258 1811 1 ! Get line from SYS$COMMAND
259 1812 1 ASSIGN_INPUT_CHANNEL, ! Assign channel to input device
260 1813 1 ASSIGN_OUTPUT_CHANNEL, ! Assign channel to output device
261 1814 1 CREATE_DIR, ! Create a directory
262 1815 1 CRYPTO_DECR_BLOCK:
263 1816 1 NOVALUE WEAK, ! Decrypt one block
264 1817 1 CRYPTO_INIDEC: WEAK, ! Setup for decryption
265 1818 1 CRYPTO_CHKSAV, ! Check for encrypted save set
266 1819 1 SYS$SETDFPROT, ! Get process default protection
267 1820 1 DEBLOCK: NOVALUE, ! Deblock save set buffer
268 1821 1 FILE_ERROR: NOVALUE, ! Signal file-related error
269 1822 1 FIND_BADBLOCK, ! Search for LBN in bad block table
270 1823 1 FREE_BUFFER: NOVALUE, ! Free buffer
271 1824 1 GET_BADBLOCKS, ! Get bad block information
```

RESTORE
V04-000

Restore a save set

K 4
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 6
(2)

:	272	1825	1	GET_BUFFER,	!	Get buffer
:	273	1826	1	GET_SUMMARY_ATTRIBUTES:	!	
:	274	1827	1	NOVALUE,	!	Get backup summary attributes
:	275	1828	1	GET_VOLUME_ATTRIBUTES:	!	
:	276	1829	1	NOVALUE,	!	Get volume attributes
:	277	1830	1	GET_FILE_ATTRIBUTES:	!	
:	278	1831	1	NOVALUE,	!	Get file attributes
:	279	1832	1	GET_FILE_EXT_ATTRIBUTES:	!	
:	280	1833	1	NOVALUE,	!	Get file attributes from extension record
:	281	1834	1	GET_PHYSVOL_ATTRIBUTES:	!	
:	282	1835	1	NOVALUE,	!	Get physical volume attributes
:	283	1836	1	INIT_NAMEBLOCK: NOVALUE,	!	Initialize extended name block fields
:	284	1837	1	MATCH,	!	Match file specifications
:	285	1838	1	RESTARTABLE_CONDITION,	!	Test if condition is restartable
:	286	1839	1	STA_INIVOL: NOVALUE,	!	Initialize output volume
:	287	1840	1	STA_INIT HDRS: NOVALUE,	!	Initialize output file headers
:	288	1841	1	STA_DISMOUNT_OUTPUT:	!	
:	289	1842	1	NOVALUE,	!	Finish file structure on volume
:	290	1843	1	STA_WRITEBOOT: NOVALUE,	!	Write boot file LBN into boot block
:	291	1844	1	WAIT:	!	Wait for I/O operation in buffer
:	292	1845	1	WILD_DIRECTORY: NOVALUE,	!	Generate output wildcard directory
:	293	1846	1	GET_RESTART: NOVALUE;	!	Get restart option
:	294	1847	1			
:	295	1848	1			
:	296	1849	1	EXTERNAL LITERAL		
:	297	1850	1	BACKUPS_BACNOTENC,		
:	298	1851	1	BACKUPS_FACILITY,		
:	299	1852	1	BACKUPS_OPENIN,		
:	300	1853	1	BACKUPS_OPENOUT,		
:	301	1854	1	BACKUPS_CLOSEIN,		
:	302	1855	1	BACKUPS_CLOSEOUT,		
:	303	1856	1	BACKUPS_COMPARED,		
:	304	1857	1	BACKUPS_CREATED,		
:	305	1858	1	BACKUPS_CREDIRERR,		
:	306	1859	1	BACKUPS_CREDIR,		
:	307	1860	1	BACKUPS_VBNMISSING,		
:	308	1861	1	BACKUPS_VBNERROR,		
:	309	1862	1	BACKUPS_VBNPASTE0F,		
:	310	1863	1	BACKUPS_FIDERROR,		
:	311	1864	1	BACKUPS_READATTR,		
:	312	1865	1	BACKUPS_GETCHN,		
:	313	1866	1	BACKUPS_WRITEATTR,		
:	314	1867	1	BACKUPS_READBLOCK,		
:	315	1868	1	BACKUPS_WRITEBLOCK,		
:	316	1869	1	BACKUPS_READERR,		
:	317	1870	1	BACKUPS_WRITEERR,		
:	318	1871	1	BACKUPS_FATALERR,		
:	319	1872	1	BACKUPS_VERIFYERR,		
:	320	1873	1	BACKUPS_BADDATA,		
:	321	1874	1	BACKUPS_PHYDEVGEO,		
:	322	1875	1	BACKUPS_BADBLOCK,		
:	323	1876	1	BACKUPS_EOFMISMATCH,		
:	324	1877	1	BACKUPS_CONTINUE,		
:	325	1878	1	BACKUPS_INVATTSTR,		
:	326	1879	1	BACKUPS_INVFID,		
:	327	1880	1	BACKUPS_NOFILES,		
:	328	1881	1	BACKUPS_STARTVERIFY,		

RESTORE
V04-000

Restore a save set

L 4
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 7
(2)

```
: 329      1882 1      BACKUP$_NONCONTIG,
: 330      1883 1      BACKUP$_NOSUCHARV,
: 331      1884 1      BACKUP$_INVATTVAL,
: 332      1885 1      BACKUP$_INCDELETE,
: 333      1886 1      BACKUP$_INCDELERR,
: 334      1887 1      BACKUP$_INCENTER,
: 335      1888 1      BACKUP$_INCBADDR,
: 336      1889 1      BACKUP$_ENCSAVSET;
: 337      1890 1
: 338      1891 1  G$DEFINE();          ! Define global common area
: 339      1892 1
: 340      1893 1
: 341      1894 1  BIND
: 342      1895 1      OUTPUT_FILENAME=      OUTPUT_ATTBUF[FAR_FILENAME]: BBLOCK,
: 343      1896 1      OUTPUT_RECATTR=      OUTPUT_ATTBUF[FAR_RECATTR]: BBLOCK,
: 344      1897 1      OUTPUT_FILECHAR=      OUTPUT_ATTBUF[FAR_UCHAR]: BBLOCK,
: 345      1898 1      OUTPUT_PLACEMENT=      OUTPUT_ATTBUF[FAR_PLACEMENT]: BBLOCK,
: 346      1899 1      OUTPUT_ACLSEGMENT=      OUTPUT_ATTBUF[FAR_ACLSEGMENT] : BBLOCK;
: 347      1900 1
: 348      1901 1
: 349      1902 1  OWN
: 350      1903 1      INCR_FIB:      BBLOCK[FIB$_LENGTH],          ! FIB for wildcarding
: 351      1904 1      INCR_RSA:      VECTOR[DIR$_NAME+6,BYTE],      ! Resultant string
: 352      1905 1      INCR_RSS,          ! Size of entire resultant string
: 353      1906 1      INCR_NTS,          ! Size of 'name.typ' portion
: 354      1907 1      INCR_VER,          ! Binary version number
: 355      1908 1      INCR_BAKDATE:      VECTOR[2];      ! Backup date
: 356      1909 1
: 357      1910 1
: 358      1911 1  BUILTIN
: 359      1912 1      INSQUE,
: 360      1913 1      REMQUE,
: 361      1914 1      ROT;
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

M 4
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 8
(3)

```

363 1915 1 XSBTTL 'INCREMENTAL_DELETE - delete file for incremental restore'
364 1916 1 ROUTINE INCREMENTAL_DELETE: NOVALUE=
365 1917 1
366 1918 1 ++
367 1919 1
368 1920 1 FUNCTIONAL DESCRIPTION:
369 1921 1 This routine conditionally deletes a file that is cataloged in a
370 1922 1 directory involved in a /INCREMENTAL restore.
371 1923 1
372 1924 1 INPUT PARAMETERS:
373 1925 1 NONE
374 1926 1
375 1927 1 IMPLICIT INPUTS:
376 1928 1 INCR_FIB, INCR_RSA, INCR_RSS, INCR_BAKDATE.
377 1929 1
378 1930 1 OUTPUT PARAMETERS:
379 1931 1 NONE
380 1932 1
381 1933 1 IMPLICIT OUTPUTS:
382 1934 1 NONE
383 1935 1
384 1936 1 ROUTINE VALUE:
385 1937 1 NONE
386 1938 1
387 1939 1 SIDE EFFECTS:
388 1940 1 If the backup date of the file is nonzero and is earlier than the
389 1941 1 creation date of the save set, the file is deleted.
390 1942 1
391 1943 1 --
392 1944 1
393 1945 2 BEGIN
394 1946 2 IF
395 1947 2 (.INCR_BAKDATE[0] OR .INCR_BAKDATE[1]) NEQ 0 AND
396 1948 3 (.INCR_BAKDATE[1] LSSU .COM_0_BSRDATE[1] OR
397 1949 4 (.INCR_BAKDATE[1] EQL .COM_0_BSRDATE[1] AND
398 1950 3 .INCR_BAKDATE[0] LSSU .COM_0_BSRDATE[0]))
399 1951 2 THEN
400 1952 3 BEGIN
401 1953 3 LOCAL
402 1954 3 STATUS, ! General status variable
403 1955 3 IOSB: VECTOR[4,WORD], ! I/O status block
404 1956 3 FIB_DESC: VECTOR[2], ! Descriptor for FIB
405 1957 3 FIB: BBLOCK[FIB$C_LENGTH], ! FIB
406 1958 3 FNA_DESC: VECTOR[2], ! Descriptor for file name
407 1959 3 RSA_DESC: VECTOR[2], ! Descriptor for RSA
408 1960 3 RSA: VECTOR[NAM$C_MAXRSS,BYTE]; ! Resultant string
409 1961 3
410 1962 3
411 1963 3 ! Construct a 'resultant string' by concatenating the device and directory
412 1964 3 ! specification, which is in the name block expanded string, with the
413 1965 3 ! file name string.
414 1966 3
415 1967 3 FNA_DESC[0] = .INCR_RSS;
416 1968 3 FNA_DESC[1] = INCR_RSA;
417 1969 3 RSA_DESC[0] = NAM$C_MAXRSS;
418 1970 3 RSA_DESC[1] = RSA;
419 P 1971 3 $FAD(
```

```

420 P 1972 3 $DESCRIPTOR('!AD!AS'),
421 P 1973 3 RSA_DESC,
422 P 1974 3 RSA_DESC,
423 P 1975 3 .OUTPUT_NAM[NAM$B_ESL], .OUTPUT_NAM[NAM$S_ESA],
424 1976 3 FNA_DESC);
425 1977 3
426 1978 3
427 1979 3 ! Set up for and delete the file.
428 1980 3 !
429 1981 3 FIB_DESC[0] = FIB$C_LENGTH;
430 1982 3 FIB_DESC[1] = FIB;
431 1983 3 CH$FILL(0, FIB$C_LENGTH, FIB);
432 1984 3 FIB[FIB$W_FID_NUM] = .INCR_FIB[FIB$W_FID_NUM];
433 1985 3 FIB[FIB$W_FID_SEQ] = .INCR_FIB[FIB$W_FID_SEQ];
434 1986 3 FIB[FIB$W_FID_RVN] = .INCR_FIB[FIB$W_FID_RVN];
435 1987 3 FIB[FIB$W_DID_NUM] = .INCR_FIB[FIB$W_DID_NUM];
436 1988 3 FIB[FIB$W_DID_SEQ] = .INCR_FIB[FIB$W_DID_SEQ];
437 1989 3 FIB[FIB$W_DID_RVN] = .INCR_FIB[FIB$W_DID_RVN];
438 P 1990 3 STATUS = $QIOQ(
439 P 1991 3 FUNC=IOS_DELETE OR IOSM_DELETE,
440 P 1992 3 CHAN=.OUTPUT_CHAN,
441 P 1993 3 IOSB=IOSB,
442 P 1994 3 P1=FIB_DESC,
443 1995 3 P2=FNA_DESC);
444 1996 3 IF .STATUS THEN STATUS = .IOSB[0];
445 1997 3
446 1998 3
447 1999 3 ! If warranted, report an error or log the deletion if requested.
448 2000 3 !
449 2001 3 IF NOT .STATUS
450 2002 3 THEN
451 2003 3 BEGIN
452 2004 3 IF .STATUS NEQ SSS_DIRNOTEMPTY
453 2005 3 THEN SIGNAL(BACKUP$_INCDELERR, 1, RSA_DESC, .STATUS);
454 2006 3 END
455 2007 3 ELSE
456 2008 3 IF .QUAL[QUAL_LOG]
457 2009 3 THEN
458 2010 3 SIGNAL(BACKUP$_INCDELETE, 1, RSA_DESC);
459 2011 3 END;
460 2012 1 END;
```

```
.TITLE RESTORE Restore a save set
.IDENT \V04-000\
```

```
.PSECT COMMON,NOEXE, OVR,2
```

```
00000 GLOBAL_BASE:
      .BLKB 0
00000 FREE_LIST:
      .BLKB 8
00008 INPUT_WAIT:
      .BLKB 8
00010 REREAD_WAIT:
      .BLKB 8
00018 OUTPUT_WAIT:
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

B 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 10
(3)

	.BLKB	8
00020 JPI_UIC:	.BLKB	4
00024 JPI_USERNAME:		
	.BLKB	12
00030 JPI_DATE:		
	.BLKB	8
00038 JPI_NODE_DESC:		
	.BLKB	8
00040 JPI_CURPRIV:		
	.BLKB	8
00048 SYI_VERSION:		
	.BLKB	4
0004C SYI_SID:	.BLKB	4
00050 RWSV_HOLD_LIST:		
	.BLKB	8
00058 RWSV_CRC16:		
	.BLKB	64
00098 RWSV_AUTODIN:		
	.BLKB	64
000D8 RWSV_FILESET_ID:		
	.BLKB	8
000E0 RWSV_VOLUME_ID:		
	.BLKB	12
000EC RWSV_VOL_NUMBER:		
	.BLKB	2
000EE RWSV_SEG_NUMBER:		
	.BLKB	2
000F0 RWSV_FILE_NUMBER:		
	.BLKB	4
000F4 RWSV_SAVE_QUAL:		
	.BLKB	4
000F8 RWSV_SAVE_FAB:		
	.BLKB	4
000FC RWSV_CHAN:		
	.BLKB	4
00100 RWSV_XOR_BCB:		
	.BLKB	4
00104 RWSV_IN_SEQ:		
	.BLKB	4
00108 RWSV_IN_SEQ 0:		
	.BLKB	4
0010C RWSV_IN_XOR_SEQ:		
	.BLKB	4
00110 RWSV_IN_XOR_RFA:		
	.BLKB	6
00116 RWSV_LOOKAHEAD:		
	.BLKB	1
00117 RWSV_XOR_SIZE:		
	.BLKB	1
00118 RWSV_IN_GROUP_SIZE:		
	.BLKB	4
0011C RWSV_IN_ERRORS:		
	.BLKB	2
0011E RWSV_IN_XORUSE:		
	.BLKB	2
00120 RWSV_IN_ORGERR:		
	.BLKB	8

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

C 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 11
(3)

00128 RWSV_IN_VBN:
 .BLKB 4
0012C RWSV_IN_VBN_0:
 .BLKB 4
00130 RWSV_ALLOC:
 .BLKB 4
00134 RWSV_EOF:
 .BLKB 4
00138 RWSV_OUT_SEQ:
 .BLKB 4
0013C RWSV_OUT_VBN:
 .BLKB 4
00140 RWSV_OUT_BLOCK_COUNT:
 .BLKB 4
00144 RWSV_OUT_ERRORS:
 .BLKB 2
00146 RWSV_SEQ_ERRORS:
 .BLKB 2
00148 RWSV_OUT_GROUP_COUNT:
 .BLKB 1
00149 RWSV_PADDING:
 .BLKB 3
0014C QUAL: .BLKB 112
001BC COM_SSNAME:
 .BLKB 8
001C4 COM_VALID_TYPES:
 .BLKB 2
001C6 COM_FLAGS:
 .BLKB 2
001C8 COM_PADDING:
 .BLKB 1
001C9 COM_BUFF_COUNT:
 .BLKB 1
001CA COM_I_SETCOUNT:
 .BLKB 1
001CB COM_O_SETCOUNT:
 .BLKB 1
001CC COM_I_STRUCNAME:
 .BLKB 12
001D8 COM_O_STRUCNAME:
 .BLKB 12
001E4 COM_O_BSRDATE:
 .BLKB 8
001EC ALT_SSNAME:
 .BLKB 32
0020C INPUT_FUNC:
 .BLKB 1
0020D INPUT_RTYPE:
 .BLKB 1
0020E OUTPUT_FUNC:
 .BLKB 1
0020F FAST_STRUCLEV:
 .BLKB 1
00210 INPUT_BEG:
 .BLKB 0
00210 INPUT_CHAN:
 .BLKB 4

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

D 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 12
(3)

00214 INPUT_FLAGS:
 .BLKB 2
00216 INPUT_PADDING:
 .BLKB 2
00218 INPUT_FAB:
 .BLKB 4
0021C INPUT_NAM:
 .BLKB 4
00220 INPUT_BCB:
 .BLKB 4
00224 INPUT_QUAL:
 .BLKB 4
00228 INPUT_BAD:
 .BLKB 4
0022C INPUT_BLOCK:
 .BLKB 4
00230 INPUT_MAXBLOCK:
 .BLKB 4
00234 INPUT_MEDIA_ID:
 .BLKB 4
00238 INPUT_NAMEDESC:
 .BLKB 8
00240 INPUT_STATBLK:
 .BLKB 8
00248 INPUT_HDR_BEG:
 .BLKB 0
00248 INPUT_CREDATE:
 .BLKB 8
00250 INPUT_REVDATE:
 .BLKB 8
00258 INPUT_EXPDATE:
 .BLKB 8
00260 INPUT_BAKDATE:
 .BLKB 8
00268 INPUT_FILEOWNER:
 .BLKB 4
0026C INPUT_FILECHAR:
 .BLKB 4
00270 INPUT_RECATTR:
 .BLKB 32
00290 INPUT_HDR_FND:
 .BLKB 0
00290 INPUT_END:
 .BLKB 0
00290 INPUT_PROC_LIST:
 .BLKB 4
00294 INPUT_PLACEMENT:
 .BLKB 8
0029C INPUT_VBN_LIST:
 .BLKB 8
002A4 INPUT_PLACE_LEN:
 .BLKB 2
002A6 INPUT_PADDING_2:
 .BLKB 2
002A8 OUTPUT_BEG:
 .BLKB 0
002A8 OUTPUT_CHAN:

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

E 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 13
(3)

002AC	OUTPUT_FLAGS:	.BLKB	4
002AE	OUTPUT_PADDING:	.BLKB	2
002B0	OUTPUT_FAB:	.BLKB	2
002B4	OUTPUT_NAM:	.BLKB	4
002B8	OUTPUT_BCB:	.BLKB	4
002BC	OUTPUT_QUAL:	.BLKB	4
002C0	OUTPUT_BAD:	.BLKB	4
002C4	OUTPUT_BLOCK:	.BLKB	4
002C8	OUTPUT_MAXBLOCK:	.BLKB	4
002CC	OUTPUT_DEVGEO:	.BLKB	8
002D4	OUTPUT_ATTBUF:	.BLKB	144
00364	OUTPUT_END:	.BLKB	0
00364	LIST_TOTFILES:	.BLKB	4
00368	LIST_TOTSIZE:	.BLKB	4
0036C	VERIFY_FAB:	.BLKB	4
00370	VERIFY_USE_COUNT:	.BLKB	4
00374	VERIFY_QUAL:	.BLKB	4
00378	COMPARE_BCB:	.BLKB	4
0037C	FAST_BUFFER:	.BLKB	4
00380	FAST_BUFFER_SIZE:	.BLKB	4
00384	FAST_RVN:	.BLKB	1
00385	FAST_PADDING:	.BLKB	1
00386	DIR_VERLIMIT:	.BLKB	2
00388	FAST_VOL_BEG:	.BLKB	0
00388	FAST_IMAP_SIZE:	.BLKB	4
0038C	FAST_IMAP:	.BLKB	4
00390	FAST_HDR_OFFSET:	.BLKB	4
00394	FAST_BOOT_LBN:	.BLKB	4

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

F 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 14
(3)

00398	FAST_VOL_END:	
	.BLKB	0
00398	JOUR_BUFFER:	
	.BLKB	4
0039C	JOUR_DIR:	
	.BLKB	4
003A0	JOUR_HIBLK:	
	.BLKB	4
003A4	JOUR_EFBLK:	
	.BLKB	4
003A8	JOUR_INBLK:	
	.BLKB	4
003AC	JOUR_FFBYTE:	
	.BLKB	2
003AE	JOUR_INBYTE:	
	.BLKB	2
003B0	JOUR_STRUCT_LEV:	
	.BLKB	2
003B2	JOUR_COUNT:	
	.BLKB	1
003B3	JOUR_REVERSE:	
	.BLKB	1
003B4	JOUR_EXSZ:	
	.BLKB	2
003B6	JOUR_PADDING:	
	.BLKB	2
003B8	CHKPT_HIGH_SP:	
	.BLKB	4
003BC	CHKPT_LOW_SP:	
	.BLKB	4
003C0	CHKPT_STACK:	
	.BLKB	4
003C4	CHKPT_VARS:	
	.BLKB	4
003C8	CHKPT_STATUS:	
	.BLKB	4
003CC	DIR_BEG:	
	.BLKB	0
003CC	DIR_CHAN:	
	.BLKB	4
003D0	DIR_NAM:	
	.BLKB	4
003D4	DIR_DEV_DESC:	
	.BLKB	4
003D8	DIR_SEL_DIR:	
	.BLKB	8
003E0	DIR_SEL_NTV:	
	.BLKB	8
003E8	DIR_STRUCT_LEV:	
	.BLKB	1
003E9	DIR_LEVELS:	
	.BLKB	1
003EA	DIR_FLAGS:	
	.BLKB	1
003EB	DIR_STATUS:	
	.BLKB	1
003EC	DIR_STRING:	
	.BLKB	320
0052C	DIR_STACK:	

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

G 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 15
(3)

```

                                .BLKB 612
00790 DIR_SP: .BLKB 4
00794 DIR_SEL_LATEST: .BLKB 4
                                .BLKB 4
00798 DIR_END: .BLKB 0
00798 DIR_SCANLIMIT: .BLKB 36
007BC INPUT_MTL: .BLKB 4
007C0 OUTPUT_MTL: .BLKB 4
007C4 CURRENT_MTL: .BLKB 4
007C8 CURRENT_VCB: .BLKB 4
007CC CURRENT_WCB: .BLKB 4
007D0 ACL_FIB_DESCR: .BLKB 8
007D8 ACL_FIB: .BLKB 64
00818 ACL_LENGTH: .BLKB 4
0081C ACL_BUFFER: .BLKB 4
00820 CRYP_IN_CONTEXT: .BLKB 4
00824 CRYP_OU_CONTEXT: .BLKB 4
00828 CRYP_DA_CONTEXT: .BLKB 4
0082C CRYP_DATA_ENCIV: .BLKB 3
00834 CRYP_DATA_CODE: .BLKB 4
00838 CRYP_DATA_KEY: .BLKB 8
00840 CRYP_DATA_IV: .BLKB 8
00848 CRYP_DATA_CKSM: .BLKB 4

                                .PSECT DATA,NOEXE,2
00000 INCR_FIB: .BLKB 64
00040 INCR_RSA: .BLKB 86
00096 .BLKB 2
00098 INCR_RSS: .BLKB 4
0009C INCR_NTS: .BLKB 4
000A0 INCR_VER: .BLKB 4
000A4 INCR_BAKDATE: .BLKB 8
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

H 5
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-S p-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 16
(3)

```
.PSECT CODE,NOWRT,2
53 41 21 44 41 21 00000 P.AAB: .ASCII \!AD!AS\
                                00006 .BLKB 2
                                00000006, 00008 P.AAA: .LONG 6
                                00000000, 0000C .ADDRESS P.AAB
```

```
OUTPUT_FILENAME= OUTPUT_ATTBUF
OUTPUT_RECATTR= OUTPUT_ATTBUF+16
OUTPUT_FILECHAR= OUTPUT_ATTBUF+100
OUTPUT_PLACEMENT= OUTPUT_ATTBUF+8
OUTPUT_ACLSEGMENT= OUTPUT_ATTBUF+128
.EXTRN LIB$CVT_DTB, LIB$GET_COMMAND
.EXTRN ASSIGN_INPUT_CHANNEL
.EXTRN ASSIGN_OUTPUT_CHANNEL
.EXTRN CREATE_DIR, CRYPTO_CHKSAV
.EXTRN SYS$SETDFPROT, DEBLOCK
.EXTRN FILE_ERROR, FIND_BADBLOCK
.EXTRN FREE_BUFFER, GET_BADBLOCKS
.EXTRN GET_BUFFER, GET_SUMMARY_ATTRIBUTES
.EXTRN GET_VOLUME_ATTRIBUTES
.EXTRN GET_FILE_ATTRIBUTES
.EXTRN GET_FILE_EXT_ATTRIBUTES
.EXTRN GET_PHYSVOL_ATTRIBUTES
.EXTRN INIT_NAMEBLOCK, MATCH
.EXTRN RESTARTABLE_CONDITION
.EXTRN STA_INIVOL, STA_INIT_HDRS
.EXTRN STA_DISMOUNT_OUTPUT
.EXTRN STA_WRITEBOOT, WAIT
.EXTRN WILD_DIRECTORY, GET_RESTART
.EXTRN BACKUP$_BACNOTENC
.EXTRN BACKUP$_FACILITY
.EXTRN BACKUP$_OPENIN, BACKUP$_OPENOUT
.EXTRN BACKUP$_CLOSEIN
.EXTRN BACKUP$_CLOSEOUT
.EXTRN BACKUP$_COMPARED
.EXTRN BACKUP$_CREATED
.EXTRN BACKUP$_CREDIRERR
.EXTRN BACKUP$_CREDIR, BACKUP$_VBNMISSING
.EXTRN BACKUP$_VBNERROR
.EXTRN BACKUP$_VBNPASTEOF
.EXTRN BACKUP$_FIDERROR
.EXTRN BACKUP$_READATTR
.EXTRN BACKUP$_GETCHN, BACKUP$_WRITEATTR
.EXTRN BACKUP$_READBLOCK
.EXTRN BACKUP$_WRITEBLOCK
.EXTRN BACKUP$_READERR
.EXTRN BACKUP$_WRITEERR
.EXTRN BACKUP$_FATALERR
.EXTRN BACKUP$_VERIFYERR
.EXTRN BACKUP$_BADDATA
.EXTRN BACKUP$_PHYDEVGEO
.EXTRN BACKUP$_BADBLOCK
.EXTRN BACKUP$_EOFMISMATCH
.EXTRN BACKUP$_CONTINUE
.EXTRN BACKUP$_INVATTSTR
.EXTRN BACKUP$_INVID, BACKUP$_NOFILES
```

.EXTRN BACKUP\$_STARTVERIFY
.EXTRN BACKUP\$_NONCONTIG
.EXTRN BACKUP\$_NOSUCHRVN
.EXTRN BACKUP\$_INVATTVAL
.EXTRN BACKUP\$_INCDELETE
.EXTRN BACKUP\$_INCDELERR
.EXTRN BACKUP\$_INCENTERR
.EXTRN BACKUP\$_INCBADDIR
.EXTRN BACKUP\$_ENCSAVSET
.EXTRN SYSS\$FAO, SYSS\$QIOW
.WEAK CRYPTO_DECR_BLOCK
.WEAK CRYPTO_INIDEC

01FC 00000 INCREMENTAL DELETE:

50	FC	A6	58 00000000G	00 9E 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	: 1916
			57 00000000'	EF 9E 00009	MOVAB	LIB\$SIGNAL, R8	
			56 00000000'	EF 9E 00010	MOVAB	COM_O_BSRDATE+4, R7	
			5E FEA0	CE 9E 00017	MOVAB	INCR_BAKDATE+4, R6	
				66 C9 0001C	MOVAB	-352(SP), SP	
				01 12 00021	BISL3	INCR_BAKDATE+4, INCR_BAKDATE, R0	: 1947
				04 00023	BNEQ	1\$	
		67		66 D1 00024	RET		
				0B 1F 00027	CMPL	INCR_BAKDATE+4, COM_O_BSRDATE+4	: 1948
				01 13 00029	BLSSU	3\$	
				04 0002B	BEQL	2\$	: 1949
				01 1F 00031	RET		
	FC	A7	FC A6 D1 0002C	2\$:	CMPL	INCR_BAKDATE, COM_O_BSRDATE	: 1950
			01 1F 00031	3\$:	BLSSU	3\$	
				04 00033	RET		
	A8	AD	F0 A6 D0 00034		MOVL	INCR_RSS, FNA_DESC	: 1967
	AC	AD	98 A6 9E 00039		MOVAB	INCR_RSA, FNA_DESC+4	: 1968
	A0	AD	FF 8F 9A 0003E		MOVZBL	#255, RSA_DESC	: 1969
	A4	AD	6E 9E 00043		MOVAB	RSA, RSA_DESC+4	: 1970
			A8 AD 9F 00047		PUSHAB	FNA_DESC	: 1976
		50	00CC C7 D0 0004A		MOVL	OUTPUT_NAM, R0	
			0C A0 DD 0004F		PUSHL	12(R0)	
		7E	0B A0 9A 00052		MOVZBL	11(R0), -(SP)	
			A0 AD 9F 00056		PUSHAB	RSA_DESC	
			A0 AD 9F 00059		PUSHAB	RSA_DESC	
			99 AF 9F 0005C		PUSHAB	P.AXA	
			00 06 FB 0005F		CALLS	#6, SYSS\$FAO	
			F0 AD 40 8F 9A 00066		MOVZBL	#64, FIB_DESC	: 1981
			F4 AD B0 AD 9E 0006B		MOVAB	FIB, FIB_DESC+4	: 1982
0040	8F		00 00 2C 00070		MOVCS	#0, (SP), #0, #64, FIB	: 1983
			B0 AD 00077				
			B4 AD FF5C C6 7D 00079		MOVQ	INCR_FIB+4, FIB+4	: 1984
			BC AD FF64 C6 D0 0007F		MOVL	INCR_FIB+12, FIB+12	: 1988
				7E 7C 00085	CLRQ	-(SP)	: 1995
				7E 7C 00087	CLRQ	-(SP)	
			A8 AD 9F 00089		PUSHAB	FNA_DESC	
			F0 AD 9F 0008C		PUSHAB	FIB_DESC	
				7E 7C 0008F	CLRQ	-(SP)	
			F8 AD 9F 00091		PUSHAB	IOSB	
		7E	0135 8F 3C 00094		MOVZWL	#309, -(SP)	
			00C0 C7 DD 00099		PUSHL	OUTPUT_CHAN	
				7E D4 0009D	CLRL	-(SP)	
			00 0C FB 0009F		CALLS	#12, SYSS\$QIOW	

RESTORE
V04-000

Restore a save set
INCREMENTAL_DELETE - delete file for incrementa

J 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 18
(3)

07		50	E9	000A6	BLBC	STATUS, 4\$	
50	F8	AD	3C	000A9	MOVZWL	IOSB, STATUS	
1A		50	E8	000AD	BLBS	STATUS, 5\$	
00002174	8F	50	D1	000B0	CMPL	STATUS, #8564	
		25	13	000B7	BEQL	6\$	
		50	DD	000B9	PUSHL	STATUS	
	A0	AD	9F	000BB	PUSHAB	RSA_DESC	
		01	DD	000BE	PUSHL	#1	
	00000000G	8F	DD	000C0	PUSHL	#BACKUP\$ INCDELERR	
	68	04	FB	000C6	CALLS	#4, LIB\$SIGNAL	
		04	000C9		RET		
OE	FF6F	C7	01	E1	000CA	5\$: BBC	#1, QUAL+11, 6\$
		A0	AD	9F	000D0	PUSHAB	RSA_DESC
		01	DD	000D3	PUSHL	#1	
	00000000G	8F	DD	000D5	PUSHL	#BACKUP\$ INCDELETE	
	68	03	FB	000DB	CALLS	#3, LIB\$SIGNAL	
		04	000DE	6\$: RET			

: 1996
: 2001
: 2004
: 2005
: 2001
: 2008
: 2010
: 2012

; Routine Size: 223 bytes, Routine Base: CODE + 0010

RESTORE
V04-000

Restore a save set

INCREMENTAL_SCAN - scan directory for increment

K 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 19
(4)

```

462 2013 1 %SBTTL 'INCREMENTAL_SCAN - scan directory for incremental restore'
463 2014 1 ROUTINE INCREMENTAL_SCAN: NOVALUE=
464 2015 1
465 2016 1 ++
466 2017 1
467 2018 1 FUNCTIONAL DESCRIPTION:
468 2019 1 This routine executes a wildcard IOS_ACCESS on a directory involved
469 2020 1 in a /INCREMENTAL restore.
470 2021 1
471 2022 1 INPUT PARAMETERS:
472 2023 1 NONE
473 2024 1
474 2025 1 IMPLICIT INPUTS:
475 2026 1 INCR_FIB, INCR_RSA, INCR_RSS.
476 2027 1
477 2028 1 OUTPUT PARAMETERS:
478 2029 1 NONE
479 2030 1
480 2031 1 IMPLICIT OUTPUTS:
481 2032 1 INCR_FIB, INCR_RSA, INCR_RSS, INCR_NTS, INCR_VER, INCR_BAKDATE.
482 2033 1
483 2034 1 ROUTINE VALUE:
484 2035 1 NONE
485 2036 1
486 2037 1 SIDE EFFECTS:
487 2038 1 NONE
488 2039 1
489 2040 1 --
490 2041 1
491 2042 2 BEGIN
492 2043 2 LOCAL
493 2044 2 FIB_DESC: VECTOR[2], Descriptor for FIB
494 2045 2 FNA_DESC: VECTOR[2], Descriptor for filename
495 2046 2 FNA: VECTOR[5,BYTE], Filename
496 2047 2 FILECHAR: $BBLOCK[4], File characteristics
497 2048 2 ATR: BLOCKVECTOR[3,8,BYTE], Attribute list
498 2049 2 STATUS, General status variable
499 2050 2 IOSB: VECTOR[4,WORD]; I/O status block
500 2051 2
501 2052 2
502 2053 2 ! Init the file ID and backup date to zero for predictable values
503 2054 2 ! in cases of error.
504 2055 2 !
505 2056 2
506 2057 2 INCR_FIB[FIB$W_FID_NUM] = INCR_FIB[FIB$W_FID_SEQ] = INCR_FIB[FIB$W_FID_RVN] = 0;
507 2058 2 INCR_BAKDATE[0] = INCR_BAKDATE[1] = 0;
508 2059 2 FIB_DESC[0] = FIB$C_LENGTH;
509 2060 2 FIB_DESC[1] = INCR_FIB;
510 2061 2 FNA_DESC[0] = 5;
511 2062 2 FNA_DESC[1] = FNA;
512 2063 2 (FNA[0])<0,32> = '*.*'; FNA[4] = '*';
513 2064 2
514 2065 2 ATR[0, ATR$W_TYPE] = ATR$C_BAKDATE;
515 2066 2 ATR[0, ATR$W_SIZE] = ATR$S_BAKDATE;
516 2067 2 ATR[0, ATR$L_ADDR] = INCR_BAKDATE;
517 2068 2 ATR[1, ATR$W_TYPE] = ATR$C_UCHAR;
518 2069 2 ATR[1, ATR$W_SIZE] = ATR$S_UCHAR;
```

```
519 2070 2 ATR[1, ATR$1_ADDR] = FILECHAR;
520 2071 2 ATR[2, 0,0,32,0] = 0;
521 2072 2
522 P 2073 2 STATUS = $Q10W(
523 P 2074 2 FUNC=IOS_ACCESS,
524 P 2075 2 CHAN=.OUTPUT_CHAN,
525 P 2076 2 IOSB=IOSB,
526 P 2077 2 P1=FIB_DESC,
527 P 2078 2 P2=FNA_DESC,
528 P 2079 2 P3=INCR_RSS,
529 P 2080 2 P4=UPLIT(DIR$$_NAME+6, INCR_RSA),
530 P 2081 2 P5=ATR
531 2082 2 );
532 2083 2 IF .STATUS THEN STATUS = .IOSB[0];
533 2084 2
534 2085 2 ! If the QIO comes back with NOSUCHFILE and a non-zero file ID, we
535 2086 2 ! have hit a dangling directory entry. Treat it as an old found file.
536 2087 2
537 2088 2
538 2089 2 IF .STATUS EQL SS$ NOSUCHFILE
539 2090 3 AND (.INCR_FIB[FIB$W_FID_NUM] NEQ 0 OR .INCR_FIB[FIB$B_FID_NMX] NEQ 0)
540 2091 2 THEN
541 2092 3 BEGIN
542 2093 3 STATUS = SS$ NORMAL;
543 2094 3 INCR_BAKDATE[0] = .INCR_BAKDATE[0] + 1;
544 2095 3 END;
545 2096 2
546 2097 2 IF NOT .STATUS
547 2098 2 THEN
548 2099 3 BEGIN
549 2100 3 INCR_RSS = 0;
550 2101 3 IF .STATUS NEQ SS$_NOMOREFILES AND .STATUS NEQ SS$_NOSUCHFILE
551 2102 3 THEN
552 2103 3 FILE_ERROR(BACKUP$_READATTR, .OUTPUT_FAB, .STATUS);
553 2104 3 END
554 2105 2 ELSE
555 2106 2
556 2107 2 ! If this is a directory file, set its backup date to -1 so that the
557 2108 2 ! incremental scan won't delete it.
558 2109 2
559 2110 3 BEGIN
560 2111 3 LOCAL
561 2112 3 T;
562 2113 3
563 2114 3 IF .FILECHAR[FCH$V_DIRECTORY]
564 2115 3 THEN INCR_BAKDATE[T] = INCR_BAKDATE[0] = -1;
565 2116 3
566 2117 3 ! Pick name and type out of the resulting filename string, and
567 2118 3 ! convert the version number to binary.
568 2119 3
569 2120 3
570 2121 3 INCR_VER = 0;
571 2122 3 INCR_NTS = .INCR_RSS;
572 2123 3 T = CH$FIND_CH(.INCR_RSS, INCR_RSA, %C';');
573 2124 3 IF .T NEQ 0
574 2125 3 THEN
575 2126 4 BEGIN
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_SCAN - scan directory for increment

M 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 21
(4)

```
: 576      2127 4      INCR NTS = .T - INCR_RSA;  
: 577      2128 4      LIB$CVT_DTB(INCR_RSA+ .INCR_RSS - (.T + 1), .T + 1, INCR_VER);  
: 578      2129 3      END;  
: 579      2130 2      END;  
: 580      2131 1      END;
```

```
00000056 000EF .BLKB 1  
00000000 000F0 P.AAC: .LONG 86  
00000000 000F4 .ADDRESS INCR_RSA
```

```
000C 00000 INCREMENTAL_SCAN:  
WORD Save R2,R3  
53 00000000' EF 0E 00002 MOVAB INCR_BAKDATE, R3  
5E 3C C2 00009 SUBL2 #60, SP  
FF62 C3 D4 0000C CLRL INCR_FIB+6  
FF60 C3 B4 00010 CLRW INCR_FIB+4  
63 7C 00014 CLRQ INCR_BAKDATE  
34 AE 40 8F 9A 00016 MOVZBL #64, FIB_DESC  
38 AE FF5C C3 9E 0001B MOVAB INCR_FIB, FIB_DESC+4  
2C AE 05 D0 00021 MOVL #5, FNA_DESC  
30 AE 24 AE 9E 00025 MOVAB FNA, FNA_DESC+4  
24 AE 3B2A2E2A 8F D0 0002A MOVL #992620074, FNA  
28 AE 2A 90 00032 MOVB #42, FNA+4  
0C AE 00140008 8F D0 00036 MOVL #1310728, ATR  
10 AE 63 9E 0003E MOVAB INCR_BAKDATE, ATR+4  
14 AE 00030004 8F D0 00042 MOVL #196812, ATR+8  
18 AE 6E 9E 0004A MOVAB FILECHAR, ATR+12  
1C AE D4 0004E CLRL ATR+16  
7E D4 00051 CLRL -(SP)  
10 AE 9F 00053 PUSHAB ATR  
9F AF 9F 00056 PUSHAB P.AAC  
F4 A3 9F 00059 PUSHAB INCR_RSS  
3C AE 9F 0005C PUSHAB FNA_DESC  
48 AE 9F 0005F PUSHAB FIB_DESC  
7E 7C 00062 CLRQ -(SP)  
24 AE 9F 00064 PUSHAB IOSB  
32 DD 00067 PUSHL #50  
00000000' EF DD 00069 PUSHL OUTPUT_CHAN  
7E D4 0006F CLRL -(SP)  
00000000G 00 0C FB 00071 CALLS #12, SYSSQ10W  
04 50 E9 00078 BLBC STATUS, 1$  
00000910 50 04 AE 3C 0007B MOVZWL IOSB, STATUS  
8F 50 D1 0007F 1$: CMPL STATUS, #2320  
11 12 00086 BNEQ 3$  
FF60 C3 B5 00088 TSTW INCR_FIB+4  
06 12 0008C BNEQ 2$  
FF65 C3 95 0008E TSTB INCR_FIB+9  
05 13 00092 BEQL 3$  
50 01 D0 00094 2$: MOVL #1, STATUS  
63 D6 00097 INCL INCR_BAKDATE  
2B 50 E8 00099 3$: BLBS STATUS, 4$  
F4 A3 D4 0009C CLRL INCR_RSS  
00000930 8F 50 D1 0009F CMPL STATUS, #2352
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_SCAN - scan directory for increment

N 5
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 22
(4)

00000910	8F		66	13	000A6	BEQL	7\$	
			50	D1	000AB	CMPL	STATUS, #2320	
			5D	13	000AF	BEQL	7\$	
		00000000'	50	DD	000B1	PUSHL	STATUS	2103
		00000000G	EF	DD	000B3	PUSHL	OUTPUT_FAB	
00000000G	00		8F	DD	000B9	PUSHL	#BACKUP\$_READATTR	
			03	FB	000BF	CALLS	#3, FILE_ERROR	
				04	000C6	RET		2097
07	01	AE	05	E1	000C7	BBC	#5, FILECHAR+1, 5\$	2114
		63	01	CE	000CC	MNEGL	#1, INCR_BAKDATE	2115
	04	A3	01	CE	000CF	MNEGL	#1, INCR_BAKDATE+4	
			A3	D4	000D3	CLRL	INCR_VER	2121
		52	A3	D0	000D6	MOVL	INCR_RSS, R2	2122
		A3	52	D0	000DA	MOVL	R2, INCR_NTS	
9C	A3	52	3B	3A	000DE	LOCC	#59, R2, INCR_RSA	2123
			02	12	000E3	BNEQ	6\$	
			51	D4	000E5	CLRL	R1	
			51	D5	000E7	TSTL	T	2124
			23	13	000E9	BEQL	7\$	
		50	A3	9E	000EB	MOVAB	INCR_RSA, R0	2127
F8	A3	51	50	C3	000EF	SUBL3	R0, T, INCR_NTS	
			A3	9F	000F4	PUSHAB	INCR_VER	2128
			01	A1	9F	PUSHAB	1(T)	
		52	9C	A342	9E	MOVAB	INCR_RSA[R2], R2	
		50	01	A1	9E	MOVAB	1(T), R0	
7E		52	50	C3	00103	SUBL3	R0, R2, -(SP)	
00000000G	00		03	FB	00107	CALLS	#3, LIB\$CVT_DTB	
			04	0010E	7\$:	RET		2131

; Routine Size: 271 bytes, Routine Base: CODE + 00F8


```
582 2132 1 %SBTTL 'ASSIGN_VERIFY_CHANNEL - assign channel for verify pass'
583 2133 1 ROUTINE ASSIGN_VERIFY_CHANNEL(NEW_FILE): NOVALUE=
584 2134 1
585 2135 1 ++
586 2136 1
587 2137 1 FUNCTIONAL DESCRIPTION:
588 2138 1 This routine assigns a channel to the input disk during a save set
589 2139 1 verification pass.
590 2140 1
591 2141 1 INPUT PARAMETERS:
592 2142 1 NEW_FILE - True if a new file is being started, false if this
593 2143 1 is the initialization call.
594 2144 1
595 2145 1 IMPLICIT INPUTS:
596 2146 1 NONE
597 2147 1
598 2148 1 OUTPUT PARAMETERS:
599 2149 1 NONE
600 2150 1
601 2151 1 IMPLICIT OUTPUTS:
602 2152 1 NONE
603 2153 1
604 2154 1 ROUTINE VALUE:
605 2155 1 NONE
606 2156 1
607 2157 1 SIDE EFFECTS:
608 2158 1 NONE
609 2159 1
610 2160 1 --
611 2161 1
612 2162 2 BEGIN
613 2163 2 LOCAL
614 2164 2 STATUS; ! Status variable
615 2165 2
616 2166 2
617 2167 2 IF .NEW_FILE
618 2168 2 THEN
619 2169 3 BEGIN
620 2170 3
621 2171 3 ! A file attribute record is being processed. Determine whether a device
622 2172 3 change occurred when this file was saved, and if one did, reassign the
623 2173 3 channel.
624 2174 3
625 2175 3 IF .QUAL[QUAL OF 11] THEN RETURN;
626 2176 3 VERIFY_USE_COUNT = .VERIFY_USE_COUNT + 1;
627 2177 3 IF .VERIFY_USE_COUNT LEQ .VERIFY_QUAL[QUAL_USE_COUNT] THEN RETURN;
628 2178 3
629 2179 3 WHILE .VERIFY_USE_COUNT GTR .VERIFY_QUAL[QUAL_USE_COUNT] DO
630 2180 4 BEGIN
631 2181 4 IF .VERIFY_QUAL[QUAL_NEXT] NEQ 0 ! Protection
632 2182 4 THEN VERIFY_QUAL = .VERIFY_QUAL[QUAL_NEXT];
633 2183 4 VERIFY_USE_COUNT = 1;
634 2184 3 END;
635 2185 3 END
636 2186 2 ELSE
637 2187 3 BEGIN
638 2188 3
```

```

639 2189 3      ! Initialization. Reassign the channel that was assigned at the conclusion
640 2190 3      ! of the previous verify pass. If this is the first, assign the channel to
641 2191 3      ! the first input device, but do not count a file, since no file is being
642 2192 3      ! processed.
643 2193 3
644 2194 3      IF .VERIFY_QUAL EQL 0
645 2195 3      THEN
646 2196 4          BEGIN
647 2197 4              VERIFY_QUAL = .QUAL[QUAL_INPU_LIST];
648 2198 4              VERIFY_USE_COUNT = 0;
649 2199 3          END;
650 2200 2      END;
651 2201 2
652 2202 2
653 2203 2      ! Assign a channel to the device.
654 2204 2
655 2205 2      IF .OUTPUT_CHAN NEQ 0 THEN $DASSGN(CHAN=.OUTPUT_CHAN);
656 2206 2      STATUS = ASSIGN_INPUT_CHANNEL(VERIFY_QUAL[QUAL_DEV_DESC], OUTPUT_CHAN, 0, 0);
657 2207 2      IF NOT .STATUS
658 2208 2      THEN
659 2209 3          BEGIN
660 2210 3              SIGNAL(
661 2211 3                  BACKUP$ OPENIN + STSK$ SEVERE, 1, VERIFY_QUAL[QUAL_DEV_DESC],
662 2212 3                  .STATUS);
663 2213 2          END;
664 2214 1      END;
```

.EXTRN SYSSDASSGN

0004 00000 ASSIGN_VERIFY_CHANNEL:

		52	00000000'	EF	9E	00002	.WORD	Save R2	2133
		2A	04	AC	E9	00009	MOVAB	VERIFY_QUAL, R2	
69	FDE7	C2		06	E0	0000D	BLBC	NEW_FILE, 3\$	2167
			FC	A2	D6	00013	BBS	#6, QUAL+15, 6\$	2175
		50		62	D0	00016	INCL	VERIFY_USE_COUNT	2176
	20	A0	FC	A2	D1	00019	MOVL	VERIFY_QUAL, R0	2177
				5C	15	0001E	CMPL	VERIFY_USE_COUNT, 32(R0)	
		50		62	D0	00020	BLEQ	6\$	
	20	A0	FC	A2	D1	00023	MOVL	VERIFY_QUAL, R0	2179
				19	15	00028	CMPL	VERIFY_USE_COUNT, 32(R0)	
				60	D5	0002A	BLEQ	4\$	
				03	13	0002C	TSTL	(R0)	2181
		62		60	D0	0002E	BEQL	2\$	
	FC	A2		01	D0	00031	MOVL	(R0), VERIFY_QUAL	2182
				E9	11	00035	MOV	#1, VERIFY_USE_COUNT	2183
				62	D5	00037	BRB	1\$	2179
				08	12	00039	TSTL	VERIFY_QUAL	2194
		62	FDD8	C2	D0	0003B	BNEQ	4\$	
			FC	A2	D4	00040	MOVL	QUAL, VERIFY_QUAL	2197
		50	FF34	C2	D0	00043	CLRL	VERIFY_USE_COUNT	2198
				09	13	00048	MOVL	OUTPUT_CHAN, R0	2205
				50	DD	0004A	BEQL	5\$	
00000000G	00			01	FB	0004C	PUSHL	R0	
				7E	7C	00053	CALLS	#1, SYSSDASSGN	
							CLRB	-(SP)	2206

RESTORE
V04-000

Restore a save set
ASSIGN_VERIFY_CHANNEL - assign channel for veri

D 6
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 25
(5)

7E	00000000G	62	FF34	C2	9F	00055	PUSHAB	OUTPUT_CHAN	:
		00		10	C1	00059	ADDL3	#16, VERIFY_QUAL, -(SP)	:
		15		04	FB	0005D	CALLS	#4, ASSIGN_INPUT_CHANNEL	:
				50	E8	00064	BLBS	STATUS, 68	: 2207
7E		62		50	DD	00067	PUSHL	STATUS	: 2212
				10	C1	00069	ADDL3	#16, VERIFY_QUAL, -(SP)	: 2211
				01	DD	0006D	PUSHL	#1	:
	00000000G	00	00000000G	8F	DD	0006F	PUSHL	#BACKUP\$ OPENIN+4	:
				04	FB	00075	CALLS	#4, LIB\$SIGNAL	:
				04	0007C	68:	RET		: 2214

; Routine Size: 125 bytes, Routine Base: CODE + 0207

```

666 2215 1 %SBTTL 'COMPARE_BLOCKS - compare file blocks'
667 2216 1 ROUTINE COMPARE_BLOCKS(FBLOCK,LBLOCK): NOVALUE=
668 2217 1
669 2218 1 ++
670 2219 1
671 2220 1 FUNCTIONAL DESCRIPTION:
672 2221 1 This routine verifies blocks.
673 2222 1
674 2223 1 INPUT PARAMETERS:
675 2224 1 FBLOCK - first block number to compare.
676 2225 1 LBLOCK - Last block number to compare.
677 2226 1
678 2227 1 IMPLICIT INPUTS:
679 2228 1 NONE
680 2229 1
681 2230 1 OUTPUT PARAMETERS:
682 2231 1 NONE
683 2232 1
684 2233 1 IMPLICIT OUTPUTS:
685 2234 1 NONE
686 2235 1
687 2236 1 ROUTINE VALUE:
688 2237 1 NONE
689 2238 1
690 2239 1 SIDE EFFECTS:
691 2240 1 NONE
692 2241 1
693 2242 1 --
694 2243 1
695 2244 2 BEGIN
696 2245 2 LOCAL
697 2246 2 STARTED_BLOCK, ! Last block on which a read is started
698 2247 2 FINISHED_BLOCK, ! Last block on which a read is done
699 2248 2 BLOCKS, ! Number of block to compare
700 2249 2 RING_A: VECTOR[2], ! Buffer queue for file A
701 2250 2 RING_B: VECTOR[2]; ! Buffer queue for file B
702 2251 2
703 2252 2
704 2253 2 ! Initialize.
705 2254 2
706 2255 2 STARTED_BLOCK = FINISHED_BLOCK = .FBLOCK;
707 2256 2 BLOCKS = .BLOCK[.FREE_LIST[0], BCB_SIZE] / 512;
708 2257 2
709 2258 2
710 2259 2 ! Form two rings of buffers. Each ring contains half the buffers.
711 2260 2 ! Start a read in each buffer.
712 2261 2
713 2262 2 RING_A[0] = RING_A[1] = RING_A;
714 2263 2 RING_B[0] = RING_B[1] = RING_B;
715 2264 2 DECR I FROM .COM_BUFF_COUNT/2 TO 1 DO
716 2265 3 BEGIN
717 2266 3 LOCAL
718 2267 3 BCB: REF BBLOCK, ! Pointer to BCB
719 2268 3 N ! Number of blocks to read this pass
720 2269 3 STATUS; ! Status variable
721 2270 3
722 2271 3
```

```

723 2272 3 IF .STARTED_BLOCK GTRU .LBLOCK THEN EXITLOOP;
724 2273 3
725 2274 3
726 2275 3 ! Compute the number of blocks to transfer.
727 2276 3 !
728 2277 3 N = .BLOCKS;
729 2278 3 IF .N GTRU .LBLOCK + 1 - .STARTED_BLOCK
730 2279 3 THEN N = .LBLOCK + 1 - .STARTED_BLOCK;
731 2280 3
732 2281 3
733 2282 3 ! Start a read for file A.
734 2283 3 !
735 2284 3 BCB = GET_BUFFER();
736 P 2285 3 STATUS = $QIO(
737 P 2286 3 FUNC=.INPUT_FUNC,
738 P 2287 3 CHAN=.INPUT_CHAN,
739 P 2288 3 EFN=BCB $ READ,
740 P 2289 3 IOSB=BCB[BCB IOSB],
741 P 2290 3 P1=.BCB[BCB_BUFFER],
742 P 2291 3 P2=.N*512,
743 2292 3 P3=.STARTED_BLOCK);
744 2293 3 IF NOT .STATUS
745 2294 3 THEN
746 2295 3 FILE_ERROR(
747 2296 3 BACKUP$ READERR + STS$K_SEVERE,
748 2297 3 .INPUT_FAB,
749 2298 3 .STATUS);
750 2299 3 BCB[BCB_SUCC_ACT] = 0;
751 2300 3 BCB[BCB_FAIL_ACT] = 0;
752 2301 3 BCB[BCB_STATE] = BCB $ READ;
753 2302 3 INSQUE(.BCB, .RING_A[1]);
754 2303 3
755 2304 3
756 2305 3 ! Start a read for file B.
757 2306 3 !
758 2307 3 BCB = GET_BUFFER();
759 P 2308 3 STATUS = $QIO(
760 P 2309 3 FUNC=.INPUT_FUNC,
761 P 2310 3 CHAN=.OUTPUT_CHAN,
762 P 2311 3 EFN=BCB $ REREAD,
763 P 2312 3 IOSB=BCB[BCB IOSB],
764 P 2313 3 P1=.BCB[BCB_BUFFER],
765 P 2314 3 P2=.N*512,
766 2315 3 P3=.STARTED_BLOCK);
767 2316 3 IF NOT .STATUS
768 2317 3 THEN
769 2318 3 FILE_ERROR(
770 2319 3 BACKUP$ READERR + STS$K_SEVERE,
771 2320 3 .OUTPUT_FAB,
772 2321 3 .STATUS);
773 2322 3 BCB[BCB_SUCC_ACT] = 0;
774 2323 3 BCB[BCB_FAIL_ACT] = 0;
775 2324 3 BCB[BCB_STATE] = BCB $ REREAD;
776 2325 3 INSQUE(.BCB, .RING_B[1]);
777 2326 3
778 2327 3
779 2328 3 ! Note that these blocks have been started.
```

```
780 2329 3 !
781 2330 3 ! STARTED_BLOCK = .STARTED_BLOCK + .N;
782 2331 3 END;
783 2332 3
784 2333 3
785 2334 3 ! Loop to wait on the first read of each ring, compare it, and reissue.
786 2335 3 !
787 2336 3 UNTIL .FINISHED_BLOCK GTRU .LBLOCK DO
788 2337 3 BEGIN
789 2338 3 LOCAL
790 2339 3 N, ! Number of blocks read this pass
791 2340 3 STATUS, ! Status variable
792 2341 3 BCB_A: REF BBLOCK, ! BCB for buffer in ring A
793 2342 3 BCB_B: REF BBLOCK; ! BCB for buffer in ring B
794 2343 3
795 2344 3
796 2345 3 ! Compute the number of blocks in this transfer.
797 2346 3 !
798 2347 3 N = .BLOCKS;
799 2348 3 IF .N GTRU .LBLOCK + 1 - .FINISHED_BLOCK
800 2349 3 THEN N = .LBLOCK + 1 - .FINISHED_BLOCK;
801 2350 3
802 2351 3
803 2352 3 ! Wait on the read for file A. If there is an error, redo the read one
804 2353 3 ! block at a time reporting the errors.
805 2354 3 !
806 2355 3 REMQUE(.RING_A[0], BCB_A);
807 2356 3 WAIT(.BCB_A);
808 2357 3 IF NOT .BCB_A[BCB_IO_STATUS]
809 2358 3 THEN
810 2359 4 BEGIN
811 2360 4 INCR I FROM 0 TO .N-1 DO
812 2361 5 BEGIN
813 2362 5 STATUS = $QIOW(
814 2363 5 FUNC=.INPUT_FUNC,
815 2364 5 CHAN=.INPUT_CHAN,
816 2365 5 IOSB=BCB_A[BCB_IOSB],
817 2366 5 P1=.BCB_A[BCB_BUFFER] + .1*512,
818 2367 5 P2=512,
819 2368 5 P3=.FINISHED_BLOCK + .1);
820 2369 5 IF .STATUS THEN STATUS = .BCB_A[BCB_IO_STATUS];
821 2370 5 IF NOT .STATUS THEN IF NOT FIND_BADBLOCK(.INPUT_BAD, .FINISHED_BLOCK + .1)
822 2371 5 THEN
823 2372 5 FILE_ERROR(
824 2373 5 BACKUP$ READBLOCK,
825 2374 5 .INPUT_FAB,
826 2375 5 .FINISHED_BLOCK + .1,
827 2376 5 .STATUS);
828 2377 4 END;
829 2378 3 END;
830 2379 3
831 2380 3
832 2381 3 ! Wait on the read for file B. If there is an error, redo the read one
833 2382 3 ! block at a time reporting the errors.
834 2383 3 !
835 2384 3 REMQUE(.RING_B[0], BCB_B);
836 2385 3 WAIT(.BCB_B);
```

```

837 2386 3 IF NOT .BCB_B[BCB_IO_STATUS]
838 2387 3 THEN
839 2388 3 BEGIN
840 2389 4 INCR I FROM 0 TO .N-1 DO
841 2390 5 BEGIN
842 2391 5 STATUS = C$QIOW(
843 2392 5 FUNC=.INPUT_FUNC,
844 2393 5 CHAN=.OUTPUT_CHAN,
845 2394 5 IOSB=BCB_B[BCB_IOSB],
846 2395 5 P1=.BCB_B[BCB_BUFFER] + .1*512,
847 2396 5 P2=512,
848 2397 5 P3=.FINISHED_BLOCK + .1);
849 2398 5 IF .STATUS THEN STATUS = .BCB_B[BCB_IO_STATUS];
850 2399 5 IF NOT .STATUS THEN IF NOT FIND_BADBLOCK(.OUTPUT_BAD, .FINISHED_BLOCK + .1)
851 2400 5 THEN
852 2401 5 FILE_ERROR(
853 2402 5 BACKUPS_READBLOCK,
854 2403 5 .OUTPUT_FAB,
855 2404 5 .FINISHED_BLOCK + .1,
856 2405 5 .STATUS);
857 2406 4 END;
858 2407 3 END;
859 2408 3
860 2409 3 ! Compare the data.
861 2410 3
862 2411 3 INCR I FROM 0 TO .N-1 DO
863 2412 4 BEGIN
864 2413 4 IF CH$NEQ(
865 2414 4 512, .BCB_A[BCB_BUFFER]+.1*512,
866 2415 4 512, .BCB_B[BCB_BUFFER]+.1*512)
867 2416 4 THEN
868 2417 4 IF NOT FIND_BADBLOCK(.INPUT_BAD, .FINISHED_BLOCK + .1)
869 2418 4 THEN
870 2419 4 FILE_ERROR(
871 2420 4 BACKUPS_VERIFYERR,
872 2421 4 .OUTPUT_FAB,
873 2422 4 .FINISHED_BLOCK + .1);
874 2423 3 END;
875 2424 3
876 2425 3
877 2426 3 ! Note that these blocks have been finished.
878 2427 3
879 2428 3 FINISHED_BLOCK = .FINISHED_BLOCK + .N;
880 2429 3
881 2430 3
882 2431 3 ! If we are done, free the buffers.
883 2432 3
884 2433 3 IF .STARTED_BLOCK GTRU .LBLOCK
885 2434 3 THEN
886 2435 4 BEGIN
887 2436 4 FREE_BUFFER(.BCB_A);
888 2437 4 FREE_BUFFER(.BCB_B);
889 2438 4 END
890 2439 3 ELSE
891 2440 4 BEGIN
892 2441 4
893 2442 4 ! Compute number of blocks to transfer in next pass.
```

```

894      2443  4      !
895      2444  4      N = .BLOCKS;
896      2445  4      IF .N GTRU .LBLOCK + 1 - .STARTED_BLOCK
897      2446  4          THEN N = .LBLOCK + 1 - .STARTED_BLOCK;
898      2447  4
899      2448  4
900      2449  4      ! Start the next read for file A.
901      2450  4      !
902      P 2451  4      STATUS = $QIO(
903      P 2452  4          FUNC=.INPUT_FUNC,
904      P 2453  4          CHAN=.INPUT_CHAN,
905      P 2454  4          EFN=BCB $ READ,
906      P 2455  4          IOSB=BCB A[BCB IOSB],
907      P 2456  4          P1=.BCB A[BCB_BUFFER],
908      P 2457  4          P2=.N*ST2,
909      2458  4          P3=.STARTED_BLOCK);
910      2459  4      IF NOT .STATUS
911      2460  4      THEN
912      2461  4          FILE_ERROR(
913      2462  4              BACKUP$ READERR + STS$K_SEVERE,
914      2463  4              .INPUT_FAB,
915      2464  4              .STATUS);
916      2465  4      BCB_A[BCB_SUCC_ACT] = 0;
917      2466  4      BCB_A[BCB_FAIL_ACT] = 0;
918      2467  4      BCB_A[BCB_STATE] = BCB $ READ;
919      2468  4      INSQUE(.BCB_A, .RING_A[1]);
920      2469  4
921      2470  4
922      2471  4      ! Start the next read for file B.
923      2472  4      !
924      P 2473  4      STATUS = C$QIO(
925      P 2474  4          FUNC=.INPUT_FUNC,
926      P 2475  4          CHAN=.OUTPUT_CHAN,
927      P 2476  4          EFN=BCB $ REREAD,
928      P 2477  4          IOSB=BCB B[BCB IOSB],
929      P 2478  4          P1=.BCB B[BCB_BUFFER],
930      P 2479  4          P2=.N*ST2,
931      2480  4          P3=.STARTED_BLOCK);
932      2481  4      IF NOT .STATUS
933      2482  4      THEN
934      2483  4          FILE_ERROR(
935      2484  4              BACKUP$ READERR + STS$K_SEVERE,
936      2485  4              .OUTPUT_FAB,
937      2486  4              .STATUS);
938      2487  4      BCB_B[BCB_SUCC_ACT] = 0;
939      2488  4      BCB_B[BCB_FAIL_ACT] = 0;
940      2489  4      BCB_B[BCB_STATE] = BCB $ REREAD;
941      2490  4      INSQUE(.BCB_B, .RING_B[1]);
942      2491  4
943      2492  4
944      2493  4      ! Note that these blocks have been started.
945      2494  4      !
946      2495  4      STARTED_BLOCK = .STARTED_BLOCK + .N;
947      2496  3      END;
948      2497  2      END;
949      2498  1      END;
```



```
.EXTRN SYSSQIO, STA_QIO
.EXTRN STA_QIOW

OFFC 00000 COMPARE_BLOCKS:
        .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
        SUBL2 #16, SP
        MOVL FBLOCK, FINISHED_BLOCK
        MOVL FINISHED_BLOCK, STARTED_BLOCK
        MOVL FREE_LIST, R0
        MOVZWL 8(R0), BLOCKS
        DIVL2 #512, BLOCKS
        MOVAB RING_A, R0
        MOVL R0, RING_A+4
        MOVL R0, RING_A
        MOVAB RING_B, R0
        MOVL R0, RING_B+4
        MOVL R0, RING_B
        MOVZBL COM_BUFF_COUNT, R7
        DIVL2 #2, R7
        INCL 1
        BRW 8$
        CMPL STARTED_BLOCK, LBLOCK
        BLEQU 2$
        BRW 10$
        MOVL BLOCKS, N
        SUBL3 STARTED_BLOCK, LBLOCK, R0
        INCL R0
        CMPL N, R0
        BLEQU 3$
        MOVL R0, N
        CALLS #0, GET_BUFFER
        MOVL R0, BCB
        CLRQ -(SP)
        CLRL -(SP)
        PUSHL STARTED_BLOCK
        ASHL #9, N, -(SP)
        PUSHL 12(BCB)
        CLRQ -(SP)
        PUSHAB 24(BCB)
        MOVZBL INPUT_FUNC, -(SP)
        PUSHL INPUT_CHAN
        PUSHL #1
        CALLS #12, SYSSQIO
        MOVL R0, STATUS
        BLBS STATUS, 4$
        PUSHL STATUS
        PUSHL INPUT_FAB
        PUSHL #BACKUP$_READERR+4
        CALLS #3, FILE_ERROR
        CLRQ 32(BCB)
        MOVAB #1, 10(BCB)
        INSQUE (BCB), @RING_A+4
        CALLS #0, GET_BUFFER
        MOVL R0, BCB
        ASHL #9, N, R4
```

5E	10	C2	00002	:	2216
55	04	AC	D0 00005	:	
5A	55	D0	00009	:	2255
50	00000000'	EF	D0 0000C	:	
5B	08	A0	3C 00013	:	2256
5B	00000200	8F	C6 00017	:	
50	08	AE	9E 0001E	:	2262
0C	AE	50	D0 00022	:	
08	AE	50	D0 00026	:	
50	6E	9E	0002A	:	2263
04	AE	50	D0 0002D	:	
6E	50	D0	00031	:	
57	00000000'	EF	9A 00034	:	2264
57		02	C6 0003B	:	
		57	D6 0003E	:	
		0102	31 00040	:	
08	AC	5A	D1 0C043	1\$:	2272
		03	1B 00047	:	
		0101	31 00049	:	
	52	5B	D0 0004C	2\$:	2277
50	08	AC	5A C3 0004F	:	2278
		50	D6 00054	:	
	50	52	D1 00056	:	
		03	1B 00059	:	
	52	50	D0 0005B	:	2279
00000000G	00	00	FB 0005E	3\$:	2284
	53	50	D0 00065	:	
		7E	7C 00068	:	2292
		7E	D4 0006A	:	
	52	5A	DD 0006C	:	
7E		09	78 0006E	:	
	0C	A3	DD 00072	:	
		7E	7C 00075	:	
	18	A3	9F 00077	:	
	7E	00000000'	EF 9A 0C07A	:	
	00000000'	EF	DD 00081	:	
		01	DD 00087	:	
00000000G	00	0C	FB 00089	:	
	56	50	D0 00090	:	
	15	56	E8 00093	:	2293
		56	DD 00096	:	2298
	00000000'	EF	DD 00098	:	2297
	00000000G	8F	DD 0009E	:	2296
00000000G	00	03	FB 000A4	:	
	20	A3	7C 000AB	4\$:	2299
	0A	A3	01 90 000AE	:	2301
	0C	BE	63 0E 000B2	:	2302
00000000G	00	00	FB 000B6	:	2307
	53	50	D0 000BD	:	
54	52	09	78 000C0	:	2315

50	18	A3	9E	000C4	MOVAB	24(R3), R0	
51	00000000'	EF	D0	000C3	MOVL	OUTPUT_CHAN, R1	
0001FFFF	8F	51	D1	000C1	CMPL	R1, #131071	
		23	1F	000D6	BLSSU	5\$	
		7E	7C	000D8	CLRQ	-(SP)	
		7E	D4	000DA	CLRL	-(SP)	
	0410	8F	BB	000DC	PUSHR	#^M<R4, R10>	
	0C	A3	DD	000E0	PUSHL	12(BCB)	
		7E	7C	000E3	CLRQ	-(SP)	
		50	DD	000E5	PUSHL	R0	
	7E	00000000'	EF	9A	000E7	MOVZBL	INPUT_FUNC, -(SP)
		51	DD	000EE	PUSHL	R1	
		04	DD	000F0	PUSHL	#4	
00000000G	00	0C	FB	000F2	CALLS	#12, STA_QIO	
		21	11	000F9	BRB	6\$	
		7E	7C	000FB	CLRQ	-(SP)	
		7E	D4	000FD	CLRL	-(SP)	
	0410	8F	BB	000FF	PUSHR	#^M<R4, R10>	
	0C	A3	DD	00103	PUSHL	12(BCB)	
		7E	7C	00106	CLRQ	-(SP)	
		50	DD	00108	PUSHL	R0	
	7E	00000000'	EF	9A	0010A	MOVZBL	INPUT_FUNC, -(SP)
		51	DD	00111	PUSHL	R1	
		04	DD	00113	PUSHL	#4	
00000000G	9F	0C	FB	00115	CALLS	#12, @#SYSSQIO	
	56	50	D0	0011C	MOVL	R0, STATUS	
	15	56	E8	0011F	BLBS	STATUS, 7\$	2316
		56	DD	00122	PUSHL	STATUS	2321
	00000000'	EF	DD	00124	PUSHL	OUTPUT_FAB	2320
	00000000G	8F	DD	0012A	PUSHL	#BACKUP\$_READERR+4	2319
00000000G	00	03	FB	00130	CALLS	#3, FILE_ERROR	
		A3	7C	00137	CLRQ	32(BCB)	2322
	0A	04	90	0013A	MOVB	#4, 10(BCB)	2324
	BE	63	0E	0013E	INSQUE	(BCB), @RING B+4	2325
	5A	52	C0	00142	ADDL2	N, STARTED_BLOCK	2330
	02	57	F5	00145	SOBGR	1, 9\$	2264
		03	11	00148	BRB	10\$	
		FEF6	31	0014A	BRW	1\$	
	08	AC	55	D1	0014D	CMPL	FINISHED_BLOCK, LBLOCK
		01	1B	00151	BLEQU	11\$	2336
			04	00153	RET		
	57	5B	D0	00154	MOVL	BLOCKS, N	2347
50	08	AC	55	C3	00157	SUBL3	FINISHED_BLOCK, LBLOCK, R0
		50	D6	0015C	INCL	R0	2348
		57	D1	0015E	CMPL	N, R0	
		03	1B	00161	BLEQU	12\$	
	57	50	D0	00163	MOVL	R0, N	2349
	58	08	BE	0F	00166	REMQUE	@RING_A, BCB_A
		58	DD	0016A	PUSHL	BCB_A	2355
00000000G	00	01	FB	0016C	CALLS	#1, WAIT	2356
	70	A8	E8	00173	BLBS	24(BCB_A), 16\$	2357
	52	01	CE	00177	MNEGL	#1, I	2368
		67	11	0017A	BRB	15\$	
		7E	7C	0017C	CLRQ	-(SP)	
		7E	D4	0017E	CLRL	-(SP)	
		6245	9F	00180	PUSHAB	(I)[FINISHED_BLOCK]	
	7E	0200	8F	3C	00183	MOVZWL	#512, -(SP)

50	52	0C B840	09 78 00188	ASHL	#9, I, R0	
		7E	9F 0018C	PUSHAB	@12(BCB_A)[R0]	
		18	7C 00190	CLRQ	-(SP)	
	7E	00000000'	9F 00192	PUSHAB	24(BCB_A)	
		00000000'	EF 9A 00195	MOVZBL	INPUT_FUNC, -(SP)	
			7E DD 0019C	PUSHL	INPUT_CHAN	
			0C FB 001A2	CLRL	-(SP)	
00000000G	00		50 DO 001A4	CALLS	#12, SYSSQIOW	
	59		59 E9 001AB	MOVL	R0, STATUS	
	07		59 E9 001AE	BLBC	STATUS, 14\$	2369
	59	18	3C 001B1	MOVZWL	24(BCB_A), STATUS	
	2B		59 E8 001B5	BLBS	STATUS, 15\$	2370
		00000000'	6245 9F 001B8 14\$:	PUSHAB	(1)[FINISHED_BLOCK]	
			EF DD 001BB	PUSHL	INPUT_BAD	
00000000G	00		02 FB 001C1	CALLS	#2, FIND_BADBLOCK	
	18		50 E8 001C8	BLBS	R0, 15\$	
			59 DD 001CB	PUSHL	STATUS	2376
		00000000'	6245 9F 001CD	PUSHAB	(1)[FINISHED_BLOCK]	2375
		00000000G	EF DD 001D0	PUSHL	INPUT_FAB	2374
			8F DD 001D6	PUSHL	#BACKOPS_READBLOCK	2372
00000000G	00		04 FB 001DC	CALLS	#4, FILE_ERROR	
95	52		57 F2 001E3 15\$:	AOBLSS	N, 1, 13\$	2360
	54	00	BE 0F 001E7 16\$:	REMQUE	@RING_B, BCB_B	2384
			54 DD 001EB	PUSHL	BCB_B	2385
00000000G	00		01 FB 001ED	CALLS	#1, WAIT	
	03	18	A4 E9 001F4	BLBC	24(BCB_B), 17\$	2386
			00AB 31 001F8	BRW	24\$	
	52		01 CE 001FB 17\$:	MNEGL	#1, I	2397
		009C	31 001FE	BRW	22\$	
53	55		52 C1 00201 18\$:	ADDL3	I, FINISHED_BLOCK, R3	
50	52		09 78 00205	ASHL	#9, I, R0	
	51	00000000'	EF DD 00209	MOVL	OUTPUT_CHAN, R1	
0001FFFF	8F		51 D1 00210	CMPL	R1, #131071	
			28 1F 00217	BLSSU	19\$	
			7E 7C 00219	CLRQ	-(SP)	
			7E D4 0021B	CLRL	-(SP)	
			53 DD 0021D	PUSHL	R3	
	7E	0200	8F 3C 0021F	MOVZWL	#512, -(SP)	
		0C B440	9F 00224	PUSHAB	@12(BCB_B)[R0]	
			7E 7C 00228	CLRQ	-(SP)	
		18	A4 9F 0022A	PUSHAB	24(BCB_B)	
	7E	00000000'	EF 9A 0022D	MOVZBL	INPUT_FUNC, -(SP)	
			51 DD 00234	PUSHL	R1	
			7E D4 00236	CLRL	-(SP)	
00000000G	00		0C FB 00238	CALLS	#12, STA_QIOW	
			26 11 0023F	BRB	20\$	
			7E 7C 00241 19\$:	CLRQ	-(SP)	
			7E D4 00243	CLRL	-(SP)	
			53 DD 00245	PUSHL	R3	
	7E	0200	8F 3C 00247	MOVZWL	#512, -(SP)	
		0C B440	9F 0024C	PUSHAB	@12(BCB_B)[R0]	
			7E 7C 00250	CLRQ	-(SP)	
		18	A4 9F 00252	PUSHAB	24(BCB_B)	
	7E	00000000'	EF 9A 00255	MOVZBL	INPUT_FUNC, -(SP)	
			51 DD 0025C	PUSHL	R1	
			7E D4 0025E	CLRL	-(SP)	
00000000G	9F		0C FB 00260	CALLS	#12, @SYSSQIOW	

59	50	D0	00267	20\$:	MOVL	R0, STATUS	2398
07	59	E9	0026A		BLBC	STATUS, 21\$	2398
59	18	A4	3C 0026D		MOVZWL	24(BCB_B), STATUS	2399
29	59	E8	00271		BLBS	STATUS, 22\$	2399
	53	DD	00274	21\$:	PUSHL	R3	2399
	EF	DD	00276		PUSHL	OUTPUT_BAD	2404
00000000G	00	02	FB 0027C		CALLS	#2, FIND_BADBLOCK	2403
	17	50	E8 00283		BLBS	R0, 22\$	2401
	0208	8F	BB 00286		PUSHR	#M<R3,R9>	2401
	00000000'	EF	DD 0028A		PUSHL	OUTPUT_FAB	2401
	00000000G	8F	DD 00290		PUSHL	#BACKUP\$_READBLOCK	2401
00000000G	00	04	FB 00296		CALLS	#4, FILE_ERROR	2389
02	52	57	F2 0029D	22\$:	AOBLSS	N, 1, 23\$	2389
		03	11 002A1		BRB	24\$	2415
		FF5B	31 002A3	23\$:	BRW	18\$	2415
	56	01	CE 002A6	24\$:	MNEGL	#1, 1	2415
		39	11 002A9		BRB	26\$	2414
50	56	09	78 002AB	25\$:	ASHL	#9, 1, R0	2414
OC B440	OC B840	8F	29 002AF		CMPC3	#512, @12(BCB_A)[R0], @12(BCB_B)[R0]	2413
	0200	29	13 002B9		BEQL	26\$	2417
		6645	9F 002BB		PUSHAB	(1)[FINISHED_BLOCK]	2417
	00000000'	EF	DD 002BE		PUSHL	INPUT_BAD	2422
00000000G	00	02	FB 002C4		CALLS	#2, FIND_BADBLOCK	2421
	16	50	E8 002CB		BLBS	R0, 26\$	2419
		6645	9F 002CE		PUSHAB	(1)[FINISHED_BLOCK]	2419
	00000000'	EF	DD 002D1		PUSHL	OUTPUT_FAB	2411
	00000000G	8F	DD 002D7		PUSHL	#BACKUP\$_VERIFYERR	2428
00000000G	00	03	FB 002DD		CALLS	#3, FILE_ERROR	2433
C3	56	57	F2 002E4	26\$:	AOBLSS	N, 1, 25\$	2436
	55	57	C0 002E8		ADDL2	N, FINISHED_BLOCK	2437
08	AC	5A	D1 002EB		CMPL	STARTED_BLOCK, LBLOCK	2433
		15	1B 002EF		BLEQU	27\$	2436
		58	DD 002F1		PUSHL	BCB_A	2437
00000000G	00	01	FB 002F3		CALLS	#1, FREE_BUFFER	2433
		54	DD 002FA		PUSHL	BCB_B	2433
00000000G	00	01	FB 002FC		CALLS	#1, FREE_BUFFER	2444
		FE47	31 00303		BRW	10\$	2445
	57	5B	D0 00306	27\$:	MOVL	BLOCKS, N	2446
50	08	5A	C3 00309		SUBL3	STARTED_BLOCK, LBLOCK, R0	2446
		50	D6 0030E		INCL	R0	2446
	50	57	D1 00310		CMPL	N, R0	2446
		03	1B 00313		BLEQU	28\$	2458
	57	50	D0 00315		MOVL	R0, N	2458
		7E	7C 00318	28\$:	CLRQ	-(SP)	2458
		7E	D4 0031A		CLRL	-(SP)	2458
		5A	DD 0031C		PUSHL	STARTED_BLOCK	2458
7E	57	09	78 0031E		ASHL	#9, N, -(SP)	2458
		OC	A8 DD 00322		PUSHL	12(BCB_A)	2458
		7E	7C 00325		CLRQ	-(SP)	2458
	18	A8	9F 00327		PUSHAB	24(BCB_A)	2458
	7E	EF	9A 0032A		MOVZBL	INPUT_FUNC, -(SP)	2458
	00000000'	EF	DD 00331		PUSHL	INPUT_CHAN	2458
	00000000G	01	DD 00337		PUSHL	#1	2458
00000000G	00	OC	FB 00339		CALLS	#12, SYS\$QIO	2459
	59	50	D0 00340		MOVL	R0, STATUS	2464
	15	59	E8 00343		BLBS	STATUS, 29\$	2464
		59	DD 00346		PUSHL	STATUS	2464

RESTORE
V04-000

Restore a save set
COMPARE_BLOCKS - compare file blocks

N 6
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 35
(6)

		00000000'	EF	DD	00348	PUSHL	INPUT_FAB	:	2463	
		00000000G	8F	DD	0034E	PUSHL	#BACKUP\$_READERR+4	:	2462	
	00000000G	00	03	FB	00354	CALLS	#3, FILE_ERROR	:		
		20	A8	7C	0035B	CLRQ	32(BCB_A)	:	2465	
	0A	A8	01	90	0035E	MOVB	#1, 107(BCB_A)	:	2467	
	0C	BE	68	0E	00362	INSQUE	(BCB_A), @RING_A+4	:	2468	
52		57	09	78	00366	ASHL	#9, R, R2	:	2480	
		18	A4	9E	0036A	MOVAB	24(R4), R1	:		
		50	00000000'	EF	D0	0036E	MOVL	OUTPUT_CHAN, R0	:	
	0001FFFF	8F	50	D1	00375	CMPL	R0, #131071	:		
			23	1F	0037C	BLSSU	30\$	:		
			7E	7C	0037E	CLRQ	-(SP)	:		
		0404	7E	D4	00380	CLRL	-(SP)	:		
		0C	8F	BB	00382	PUSHR	#^M<R2,R10>	:		
			A4	DD	00386	PUSHL	12(BCB_B)	:		
			7E	7C	00389	CLRQ	-(SP)	:		
		7E	51	DD	0038B	PUSHL	R1	:		
		00000000'	EF	9A	0038D	MOVZBL	INPUT_FUNC, -(SP)	:		
			50	DD	00394	PUSHL	R0	:		
			04	DD	00396	PUSHL	#4	:		
	00000000G	00	0C	FB	00398	CALLS	#12, STA_QIO	:		
			21	11	0039F	BRB	31\$	:		
			7E	7C	003A1	CLRQ	-(SP)	:		
			7E	D4	003A3	CLRL	-(SP)	:		
		0404	8F	BB	003A5	PUSHR	#^M<R2,R10>	:		
		0C	A4	DD	003A9	PUSHL	12(BCB_B)	:		
			7E	7C	003AC	CLRQ	-(SP)	:		
		7E	51	DD	003AE	PUSHL	R1	:		
		00000000'	EF	9A	003B0	MOVZBL	INPUT_FUNC, -(SP)	:		
			50	DD	003B7	PUSHL	R0	:		
			04	DD	003B9	PUSHL	#4	:		
	00000000G	9F	0C	FB	003BB	CALLS	#12, @#SYSS\$QIO	:		
		9	50	D0	003C2	MOVL	R0, STATUS	:		
		15	59	E8	003C5	BLBS	STATUS, 32\$	:	2481	
			59	DD	003C8	PUSHL	STATUS	:	2486	
		00000000'	EF	DD	003CA	PUSHL	OUTPUT_FAB	:	2485	
		00000000G	8F	DD	003D0	PUSHL	#BACKUP\$_READERR+4	:	2484	
			03	FB	003D6	CALLS	#3, FILE_ERROR	:		
		20	A4	7C	003DD	CLRQ	32(BCB_B)	:	2487	
	0A	A4	04	90	003E0	MOVB	#4, 107(BCB_B)	:	2489	
	04	BE	64	0E	003E4	INSQUE	(BCB_B), @RING_B+4	:	2490	
		5A	57	C0	003E8	ADDL2	N, STARTED_BLOCK	:	2495	
			FD5F	31	003EB	BRW	10\$	:	2336	
			04	003EE	RET			:	2498	

; Routine Size: 1007 bytes, Routine Base: CODE + 0284

```

: 951      2499 1 %SBTTL 'GET_ACP_ATTRIBUTES - generate ACP attribute list'
: 952      2500 1 ROUTINE GET_ACP_ATTRIBUTES(ATR_LIST): NOVALUE=
: 953      2501 1
: 954      2502 1 ++
: 955      2503 1
: 956      2504 1 FUNCTIONAL DESCRIPTION:
: 957      2505 1 This routine generates the ACP attribute list.
: 958      2506 1
: 959      2507 1 INPUT PARAMETERS:
: 960      2508 1 ATR_LIST - Pointer to receiving buffer
: 961      2509 1
: 962      2510 1 IMPLICIT INPUTS:
: 963      2511 1 OUTPUT_ATTBUF - Attributes for output file.
: 964      2512 1
: 965      2513 1 OUTPUT PARAMETERS:
: 966      2514 1 NONE
: 967      2515 1
: 968      2516 1 IMPLICIT OUTPUTS:
: 969      2517 1 NONE
: 970      2518 1
: 971      2519 1 ROUTINE VALUE:
: 972      2520 1 NONE
: 973      2521 1
: 974      2522 1 SIDE EFFECTS:
: 975      2523 1 NONE
: 976      2524 1
: 977      2525 1 --
: 978      2526 1
: 979      2527 2 BEGIN
: 980      2528 2
: 981      2529 2 $ASSUME (MAX_ATTRS, GEQU, 13) ! Number of attributes generated by this routine
: 982      2530 2
: 983      2531 2 LITERAL
: 984      2532 2 ATRSS_ASCDATES = 2; ! Only write 2 bytes of ASCDATES for revision count
: 985      2533 2 MACRO
: 986      2534 2 GEN_ATTR_LIST[A,B]=
: 987      2535 2 .ATR = %NAME('ATRSS_', A) + %NAME('ATRSC_', A)^16;
: 988      2536 2 ATR = .ATR + 4;
: 989      2537 2 .ATR = B;
: 990      2538 2 ATR = .ATR + 4 %;
: 991      2539 2 LOCAL
: 992      2540 2 ATR; ! Pointer to attribute list
: 993      2541 2
: 994      2542 2
: 995      2543 2 ! Build the attribute descriptor list.
: 996      2544 2
: 997      2545 2 ATR = .ATR_LIST;
: 998      2546 2 IF NOT .QUAL[QUAL_INTE]
: 999      2547 2 OR .QUAL[QUAL_IMAG]
1000      2548 2 THEN
1001      2549 3 BEGIN
1002      2550 3 GEN_ATTR_LIST(
1003      2551 3 FPRO, OUTPUT_ATTBUF[FAR_FPRO],
1004      2552 3 RPRO, OUTPUT_ATTBUF[FAR_RPRO]);
1005      2553 2 END;
1006      2554 2 GEN_ATTR_LIST(
1007      2555 2 ACLEVEL, OUTPUT_ATTBUF[FAR_ACLEVEL],
```

```

: 1008 P 2556 2 UCHAR, OUTPUT_ATTBUF[FAR_UCHAR],
: 1009 P 2557 2 JOURNAL, OUTPUT_ATTBUF[FAR_JNL_FLAGS],
: 1010 2558 2 RECATTR, OUTPUT_ATTBUF[FAR_RECATTR]);
: 1011 2559 2 IF .(OUTPUT_ATTBUF[FAR_CREDATE]) NEQ 0
: 1012 2560 2 THEN
: 1013 2561 2 BEGIN
: 1014 P 2562 2 GEN_ATTR_LIST(
: 1015 2563 2 -CREDATE, OUTPUT_ATTBUF[FAR_CREDATE]);
: 1016 2564 2 END;
: 1017 2565 2 IF .(OUTPUT_ATTBUF[FAR_REVDATE]) NEQ 0
: 1018 2566 2 THEN
: 1019 2567 2 BEGIN
: 1020 P 2568 2 GEN_ATTR_LIST(
: 1021 P 2569 2 -REVDATE, OUTPUT_ATTBUF[FAR_REVDATE],
: 1022 2570 2 ASCDATES, OUTPUT_ATTBUF[FAR_REVISION]);
: 1023 2571 2 END;
: 1024 2572 2 IF .(OUTPUT_ATTBUF[FAR_EXPDATE]) NEQ 0
: 1025 2573 2 THEN
: 1026 2574 2 BEGIN
: 1027 P 2575 2 GEN_ATTR_LIST(
: 1028 2576 2 -EXPDATE, OUTPUT_ATTBUF[FAR_EXPDATE]);
: 1029 2577 2 END;
: 1030 2578 2 IF .QUAL[QUAL_RECO] OR .QUAL[QUAL_INCR]
: 1031 2579 2 THEN
: 1032 2580 2 BEGIN
: 1033 P 2581 2 GEN_ATTR_LIST(
: 1034 2582 2 -BAKDATE, COM_O_BSRDATE);
: 1035 2583 2 END
: 1036 2584 2 ELSE
: 1037 2585 2 IF .QUAL[QUAL_IMAG] AND .(OUTPUT_ATTBUF[FAR_BAKDATE]) NEQ 0
: 1038 2586 2 THEN
: 1039 2587 2 BEGIN
: 1040 P 2588 2 GEN_ATTR_LIST(
: 1041 2589 2 -BAKDATE, OUTPUT_ATTBUF[FAR_BAKDATE]);
: 1042 2590 2 END;
: 1043 P 2591 2 GEN_ATTR_LIST(
: 1044 2592 2 -UIC, OUTPUT_ATTBUF[FAR_UIC]);
: 1045 2593 2 .ATR = 0;
: 1046 2594 1 END;
```

0004 0000 GET_ACP_ATTRIBUTES:

		52	00000000'	EF	9E	00002		.WORD	Save R2	: 2500
		50	04	AC	D0	00009		MOVAB	QUAL+8, R2	
05	06	A2		01	E1	0000D		MOVL	ATR_LIST, ATR	: 2545
18	02	A2		03	E1	00012		BBC	#1, QUAL+14, 1\$	: 2546
		80	00160002	8F	D0	00017	1\$:	BBC	#3, QUAL+10, 2\$	: 2547
		80	01EE	C2	9E	0001E		MOVL	#1441794, (ATR)+	: 2552
		80	00170002	8F	D0	00023		MOVAB	OUTPUT_ATTBUF+110, (ATR)+	
		80	01F0	C2	9E	0002A		MOVL	#1507330, (ATR)+	
		80	00180001	8F	D0	0002F	2\$:	MOVAB	OUTPUT_ATTBUF+112, (ATR)+	
		80	01FA	C2	9E	00036		MOVL	#1572865, (ATR)+	: 2558
		80	00030004	8F	D0	0003B		MOVAB	OUTPUT_ATTBUF+122, (ATR)+	
								MOVL	#196612, (ATR)+	

RESTORE
V04-000

Restore a save set
GET_ACP_ATTRIBUTES - generate ACP attribute lis

D 7
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 38
(7)

		80	01E4	C2	9E	0C042	MOVAB	OUTPUT_ATTBUF+100, (ATR)+	:
		80	001D0002	8F	D0	00047	MOVL	#1900526, (ATR)+	:
		80	020C	C2	9E	0004E	MOVAB	OUTPUT_ATTBUF+140, (ATR)+	:
		80	00040020	8F	D0	00053	MOVL	#262178, (ATR)+	:
		80	0190	C2	9E	0005A	MOVAB	OUTPUT_ATTBUF+16, (ATR)+	:
			01B0	C2	D5	0005F	TSTL	OUTPUT_ATTBUF+48	:
				0C	13	00063	BEQL	3\$	2559
		80	00110008	8F	D0	00065	MOVL	#1114120, (ATR)+	2563
		80	01B0	C2	9E	0006C	MOVAB	OUTPUT_ATTBUF+48, (ATR)+	:
			01B8	C2	D5	00071	TSTL	OUTPUT_ATTBUF+56	2565
				18	13	00075	BEQL	4\$	:
		80	00120008	8F	D0	00077	MOVL	#1179656, (ATR)+	2570
		80	01B8	C2	9E	0007E	MOVAB	OUTPUT_ATTBUF+56, (ATR)+	:
		80	000D0002	8F	D0	00083	MOVL	#851970, (ATR)+	:
		80	01F2	C2	9E	0008A	MOVAB	OUTPUT_ATTBUF+114, (ATR)+	:
			01C0	C2	D5	0008F	TSTL	OUTPUT_ATTBUF+64	2572
				0C	13	00093	BEQL	5\$	:
		80	00130008	8F	D0	00095	MOVL	#1245192, (ATR)+	2576
		80	01C0	C2	9E	0009C	MOVAB	OUTPUT_ATTBUF+64, (ATR)+	:
05	04	A2		06	E0	000A1	BBS	#6, QUAL+12, 6\$	2578
0E	02	A2		04	E1	000A6	BBC	#4, QUAL+10, 7\$	:
		80	00140008	8F	D0	000AB	MOVL	#1310728, (ATR)+	2582
		60	0090	C2	9E	000B2	MOVAB	COM_O_BSRDATE, (ATR)	:
				17	11	000B7	BRB	8\$	:
15	02	A2		03	E1	000B9	BBC	#3, QUAL+10, 9\$	2585
			01C8	C2	D5	000BE	TSTL	OUTPUT_ATTBUF+72	:
				0F	13	000C2	BEQL	9\$	:
		80	00140008	8F	D0	000C4	MOVL	#1310728, (ATR)+	2589
		60	01C8	C2	9E	000CB	MOVAB	OUTPUT_ATTBUF+72, (ATR)	:
		50		04	C0	000D0	ADDL2	#4, ATR	:
		80	00150004	8F	D0	000D3	MOVL	#1376260, (ATR)+	2592
		80	01E0	C2	9E	000DA	MOVAB	OUTPUT_ATTBUF+96, (ATR)+	:
				60	D4	000DF	CLRL	(ATR)	2593
				04	000E1		RET		2594

; Routine Size: 226 bytes, Routine Base: CODE + 0673


```
1048 2595 1 %SBTTL 'GET_PARENT_UIC - get parent UIC of file'
1049 2596 1 ROUTINE GET_PARENT_UIC=
1050 2597 1
1051 2598 1 ++
1052 2599 1
1053 2600 1 FUNCTIONAL DESCRIPTION:
1054 2601 1 This routine gets the parent UIC.
1055 2602 1
1056 2603 1 INPUT PARAMETERS:
1057 2604 1 NONE
1058 2605 1
1059 2606 1 IMPLICIT INPUTS:
1060 2607 1 NONE
1061 2608 1
1062 2609 1 OUTPUT PARAMETERS:
1063 2610 1 NONE
1064 2611 1
1065 2612 1 IMPLICIT OUTPUTS:
1066 2613 1 NONE
1067 2614 1
1068 2615 1 ROUTINE VALUE:
1069 2616 1 The parent's UIC.
1070 2617 1
1071 2618 1 SIDE EFFECTS:
1072 2619 1 NONE
1073 2620 1
1074 2621 1 --
1075 2622 1
1076 2623 2 BEGIN
1077 2624 2 LOCAL
1078 2625 2 DEV_DESC: VECTOR[2], Device name descriptor
1079 2626 2 CHANNEL, Channel
1080 2627 2 STATUS, Status variable
1081 2628 2 IOSB: VECTOR[4,WORD], I/O status block
1082 2629 2 FIB: BBLOCK[FIB$C_LENGTH], FIB
1083 2630 2 FIB_DESC: VECTOR[2], Descriptor for FIB
1084 2631 2 UIC, Return location for UIC
1085 2632 2 ATR_DESC: BBLOCK[12]; ACP attribute list
1086 2633 2
1087 2634 2
1088 2635 2 ! Assign a channel to the device.
1089 2636 2
1090 2637 2 DEV_DESC[0] = .OUTPUT_NAM[NAM$B_DEV];
1091 2638 2 DEV_DESC[1] = .OUTPUT_NAM[NAM$L_DEV];
1092 2639 2 STATUS = ASSIGN_OUTPUT_CHANNEL(DEV_DESC, CHANNEL, 0, 0);
1093 2640 2 IF NOT .STATUS
1094 2641 2 THEN
1095 2642 2 BEGIN
1096 2643 2 FILE_ERROR(
1097 2644 2 BACKUP$.READATTR,
1098 2645 2 .OUTPUT_FAB,
1099 2646 2 .STATUS);
1100 2647 2 RETURN .JPI_UIC;
1101 2648 2 END;
1102 2649 2
1103 2650 2
1104 2651 2 ! Initialize the FIB.
```

RESTORE
V04-000

Restore a save set
GET_PARENT_UIC - get parent UIC of file

F 7
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 40
(8)

```

: 1105 2652 2 !
: 1106 2653 2 CH$FILL (0, FIB$C_LENGTH, FIB);
: 1107 2654 2 FIB[FIB$C_ACCTL] = 0;
: 1108 2655 2 FIB[FIB$C_FID_NUM] = .OUTPUT_NAM[NAM$W_DID_NUM];
: 1109 2656 2 FIB[FIB$C_FID_SEQ] = .OUTPUT_NAM[NAM$W_DID_SEQ];
: 1110 2657 2 FIB[FIB$C_FID_RVN] = .OUTPUT_NAM[NAM$W_DID_RVN];
: 1111 2658 2 FIB_DESC[0] = FIB$C_LENGTH;
: 1112 2659 2 FIB_DESC[1] = FIB;
: 1113 2660 2
: 1114 2661 2
: 1115 2662 2 ! Initialize the ACP attribute list.
: 1116 2663 2
: 1117 2664 2 ATR_DESC[0,0,16,0] = ATR$S_UIC;
: 1118 2665 2 ATR_DESC[2,0,16,0] = ATR$C_UIC;
: 1119 2666 2 ATR_DESC[4,0,32,0] = UIC;
: 1120 2667 2 ATR_DESC[8,0,32,0] = 0;
: 1121 2668 2
: 1122 2669 2
: 1123 2670 2 ! Execute the read attributes.
: 1124 2671 2
: 1125 P 2672 2 STATUS = $QIOW(
: 1126 P 2673 2     FUNC=IOS_ACCESS,
: 1127 P 2674 2     CHAN=.CHANNEL,
: 1128 P 2675 2     IOSB=IOSB,
: 1129 P 2676 2     P1=FIB_DESC,
: 1130 2677 2     P5=ATR_DESC);
: 1131 2678 2 IF .STATUS THEN STATUS = .IOSB[0];
: 1132 2679 2 IF NOT .STATUS
: 1133 2680 2 THEN
: 1134 2681 2     BEGIN
: 1135 2682 2     FILE_ERROR(
: 1136 2683 2         BACKUP$ READATTR,
: 1137 2684 2         .OUTPUT_FAB,
: 1138 2685 2         .STATUS);
: 1139 2686 2     UIC = .JPI_UIC;
: 1140 2687 2     END;
: 1141 2688 2
: 1142 2689 2
: 1143 2690 2 ! Deassign the channel.
: 1144 2691 2
: 1145 2692 2 $DASSGN(CHAN=.CHANNEL);
: 1146 2693 2
: 1147 2694 2
: 1148 2695 2 ! Return the UIC.
: 1149 2696 2
: 1150 2697 2 .UIC
: 1151 2698 2 END;
```

03FC 00000 GET_PARENT_UIC:

```

59 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9
58 00000000G 8F D0 00009 MOVAB FILE_ERROR, R9
57 00000000G EF 9E 00010 MOVL #BACKUP$ READATTR, R8
MOVAB OUTPUT_NAM, R7
```

: 2596
:
:
:

RESTORE
V04-000

Restore a save set
GET_PARENT_UIC - get parent UIC of file

G 7

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742

[BACKUP.SRC]RESTORE.B32;1

Page 41
(8)

		SE	94	AE	9E	00017	MOVAB	-108(SP), SP		
		50		67	D0	0001B	MOVL	OUTPUT_NAM, R0	2637	
	64	AE	39	A0	9A	0001E	MOVZBL	57(R0), DEV_DESC		
	68	AE	44	A0	D0	00023	MOVL	68(R0), DEV_DESC+4	2638	
				7E	7C	00028	CLRQ	-(SP)	2639	
			08	AE	9F	0002A	PUSHAB	CHANNEL		
			70	AE	9F	0002D	PUSHAB	DEV_DESC		
	00000000G	00		04	FB	00030	CALLS	#4, ASSIGN_OUTPUT_CHANNEL		
		56		50	D0	00037	MOVL	R0, STATUS		
		10		56	E8	0003A	BLBS	STATUS, 1\$	2640	
				56	DD	0003D	PUSHL	STATUS	2646	
			FC	A7	DD	0003F	PUSHL	OUTPUT_FAB	2645	
				58	DD	00042	PUSHL	R8	2643	
		69		03	FB	00044	CALLS	#3, FILE_ERROR		
		50	FD6C	C7	D0	00047	MOVL	JPI_UIC, R0	2647	
					04	0004C	RET			
0040	8F		00	6E	00	2C	0004D	1\$: MOVCS	#0, (SP), #0, #64, FIB	2653
				1C	AE		00054			
				1C	AE	D4	00056	CLRL	FIB	2654
		50		67	D0	00059	MOVL	OUTPUT_NAM, R0	2655	
	20	AE	2A	A0	D0	0005C	MOVL	42(R0), FIB+4		
	24	AE	2E	A0	B0	00061	MOVW	46(R0), FIB+8	2657	
	14	AE	40	8F	9A	00066	MOVZBL	#64, FIB_DESC	2658	
	18	AE	1C	AE	9E	0006B	MOVAB	FIB, FIB_DESC+4	2659	
	08	AE	00150004	8F	D0	00070	MOVL	#1376260, ATR_DESC	2664	
	0C	AE		04	AE	9E	00078	MOVAB	UIC, ATR_DESC+4	2666
				10	AE	D4	0007D	CLRL	ATR_DESC+8	2667
				7E	D4	00080	CLRL	-(SP)	2677	
			0C	AE	9F	00082	PUSHAB	ATR_DESC		
				7E	7C	00085	CLRQ	-(SP)		
				7E	D4	00087	CLRL	-(SP)		
			28	AE	9F	00089	PUSHAB	FIB_DESC		
				7E	7C	0008C	CLRQ	-(SP)		
			7C	AE	9F	0008E	PUSHAB	IOSB		
				32	DD	00091	PUSHL	#50		
			28	AE	DD	00093	PUSHL	CHANNEL		
				7E	D4	00096	CLRL	-(SP)		
	00000000G	00		0C	FB	00098	CALLS	#12, SYSSQIOW		
		56		50	D0	0009F	MOVL	R0, STATUS		
		07		56	E9	000A2	BLBC	STATUS, 2\$	2678	
		56	5C	AE	3C	000A5	MOVZWL	IOSB, STATUS		
		10		56	E8	000A9	BLBS	STATUS, 3\$	2679	
				56	DD	000AC	PUSHL	STATUS	2685	
			FC	A7	DD	000AE	PUSHL	OUTPUT_FAB	2684	
				58	DD	000B1	PUSHL	R8	2682	
		69		03	FB	000B3	CALLS	#3, FILE_ERROR		
	04	AE	FD6C	C7	D0	000B6	MOVL	JPI_UIC, UIC	2686	
				6E	DD	000BC	PUSHL	CHANNEL	2692	
	00000000G	00		01	FB	000BE	CALLS	#1, SYSSDASSGN		
		50		04	AE	D0	000C5	MOVL	UIC, R0	2698
					04	000C9	RET			

; Routine Size: 202 bytes, Routine Base: CODE + 0755

```
1153 2699 1 %SBTTL 'CLOSE RESTORE - close restored file'
1154 2700 1 GLOBAL ROUTINE CLOSE_RESTORE: NOVALUE=
1155 2701 1
1156 2702 1 ++
1157 2703 1
1158 2704 1 FUNCTIONAL DESCRIPTION:
1159 2705 1 This routine finishes an open output file.
1160 2706 1
1161 2707 1 INPUT PARAMETERS:
1162 2708 1 NONE
1163 2709 1
1164 2710 1 IMPLICIT INPUTS:
1165 2711 1 NONE
1166 2712 1
1167 2713 1 OUTPUT PARAMETERS:
1168 2714 1 NONE
1169 2715 1
1170 2716 1 IMPLICIT OUTPUTS:
1171 2717 1 NONE
1172 2718 1
1173 2719 1 ROUTINE VALUE:
1174 2720 1 NONE
1175 2721 1
1176 2722 1 SIDE EFFECTS:
1177 2723 1 NONE
1178 2724 1
1179 2725 1 --
1180 2726 1
1181 2727 2 BEGIN
1182 2728 2 IF .OUTPUT_FLAGS[OUTPUT_INCR_DIR]
1183 2729 2 THEN
1184 2730 3 BEGIN
1185 2731 3 LOCAL
1186 2732 3 FIB: BBLOCK[FIB$C_LENGTH], FIB
1187 2733 3 FIB_DESC: VECTOR[2], Descriptor for FIB
1188 2734 3 STATUS, Status return
1189 2735 3 IOSB: VECTOR[4,WORD]; I/O status block
1190 2736 3
1191 2737 3
1192 2738 3 ! Process files in the directory that appear after the last entry in the
1193 2739 3 ! saved copy of the directory.
1194 2740 3
1195 2741 3 WHILE .INCR_RSS NEQ 0 DO
1196 2742 4 BEGIN
1197 2743 4 INCREMENTAL_DELETE();
1198 2744 4 INCREMENTAL_SCAN();
1199 2745 3 END;
1200 2746 3
1201 2747 3
1202 2748 3 ! Update the directory's backup date.
1203 2749 3
1204 2750 3 CH$FILL(0, FIB$C_LENGTH, FIB);
1205 2751 3 FIB[FIB$C_ACCTL] = FIB$M_WRITE OR FIB$M_NORECORD;
1206 2752 3 FIB[FIB$W_FID_NUM] = .INCR_FIB[FIB$W_DID_NUM];
1207 2753 3 FIB[FIB$W_FID_SEQ] = .INCR_FIB[FIB$W_DID_SEQ];
1208 2754 3 FIB[FIB$W_FID_RVN] = .INCR_FIB[FIB$W_DID_RVN];
1209 2755 3 FIB_DESC[0] = FIB$C_LENGTH;
```

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

1 7
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 43
(9)

```

: 1210      2756      3      FIB_DESC[1] = FIB;
: 1211      2757      3      STATUS = $QIOW(
P 2758      3      FUNC=IOS_MODIFY,
P 2759      3      CHAN=.OUTPUT_CHAN,
P 2760      3      IOSB=IOSB,
P 2761      3      P1=FIB_DESC,
P 2762      3      P5=UPLIT(
P 2763      3      WORD(ATTR$ BAKDATE, ATTR$C_BAKDATE),
P 2764      3      COM_O_BSRDATE,
: 1219      2765      3      0));
: 1220      2766      3      IF .STATUS THEN STATUS = .IOSB[0];
: 1221      2767      3      IF NOT .STATUS
: 1222      2768      3      THEN
: 1223      2769      3      FILE_ERROR(
: 1224      2770      3      BACKUP$WRITEATTR,
: 1225      2771      3      .OUTPUT_FAB,
: 1226      2772      3      .STATUS);
: 1227      2773      3
: 1228      2774      3
: 1229      2775      3      OUTPUT_FLAGS[OUTPUT_INCR_DIR] = FALSE;
: 1230      2776      3      $DASSGN(CHAN=.OUTPUT_CHAN);
: 1231      2777      3      OUTPUT_CHAN = 0;
: 1232      2778      3      END
: 1233      2779      2      ELSE IF .OUTPUT_FLAGS[OUTPUT_DIR_OPEN]
: 1234      2780      2      THEN
: 1235      2781      3      BEGIN
: 1236      2782      3      $DASSGN(CHAN=.OUTPUT_CHAN);
: 1237      2783      3      OUTPUT_FLAGS[OUTPUT_DIR_OPEN] = FALSE ;
: 1238      2784      3      OUTPUT_CHAN = 0;
: 1239      2785      3      END ;
: 1240      2786      2
: 1241      2787      2      IF .OUTPUT_FLAGS[OUTPUT_OPEN]
: 1242      2788      2      THEN
: 1243      2789      3      BEGIN
: 1244      2790      3      LOCAL
: 1245      2791      3      BCB:          REF BBLOCK,      ! Pointer to BCB
: 1246      2792      3      STATUS,          ! Status variable
: 1247      2793      3      IOSB:          VECTOR[4,WORD]; ! I/O status block
: 1248      2794      3
: 1249      2795      3
: 1250      2796      3      ! Wait for all pending writes to complete.
: 1251      2797      3
: 1252      2798      3      UNTIL REMQUE(.OUTPUT_WAIT[0], BCB) DO
: 1253      2799      4      BEGIN
: 1254      2800      4      WAIT(.BCB);
: 1255      2801      4      FREE_BUFFER(.BCB);
: 1256      2802      3      END;
: 1257      2803      3
: 1258      2804      3
: 1259      2805      3      ! Make sure we got all the data records we were expecting. Accept an error
: 1260      2806      3      ! if this is the end of a tape verify pass, since the file may legitimately
: 1261      2807      3      ! be incomplete. Also, allow a number of blocks greater than what we were
: 1262      2808      3      ! expecting, since a message has already been issued in this case. Accept
: 1263      2809      3      ! an error if /VOLUME is specified. If the file was implicitly opened, we
: 1264      2810      3      ! do not know the end-of-file block, so the error must be accepted. If the
: 1265      2811      3      ! input save set is not sequential then we must accept an error also.
: 1266      2812      3
```

```
1267 2813 3 IF
1268 2814 3 NOT .QUAL[QUAL_VOLU] AND
1269 2815 3 NOT .OUTPUT_FLAGS[OUTPUT_V_LAST] AND
1270 2816 3 NOT .OUTPUT_FLAGS[OUTPUT_IMPLICIT] AND
1271 2817 3 NOT .OUTPUT_FLAGS[OUTPUT_NONSEQ_IN] AND
1272 2818 3 .OUTPUT_BLOCK LEQ .OUTPUT_MAXBLOCK AND
1273 2819 3 .OUTPUT_BLOCK NEQ 1
1274 2820 3 THEN
1275 2821 3 FILE_ERROR(
1276 2822 3 BACKUPS_VBNMISSING,
1277 2823 3 .OUTPUT_FAB,
1278 2824 3 .OUTPUT_BLOCK, .OUTPUT_MAXBLOCK);
1279 2825 3
1280 2826 3
1281 2827 3 ! If verification is enabled on a Files-11 to Files-11 operation, do it.
1282 2828 3
1283 2829 3 IF .QUAL[QUAL_IF11] AND .QUAL[QUAL_OF11] AND .QUAL[QUAL_VERI]
1284 2830 3 THEN
1285 2831 3 BEGIN
1286 2832 3
1287 2833 3 ! Log the file creation.
1288 2834 3
1289 2835 3 IF .QUAL[QUAL_LOG]
1290 2836 3 THEN
1291 2837 3 FILE_ERROR(
1292 2838 3 BACKUPS_CREATED, .OUTPUT_FAB);
1293 2839 3
1294 2840 3
1295 2841 3 ! Do the verification.
1296 2842 3
1297 2843 3 IF .QUAL[QUAL_PHYS]
1298 2844 3 THEN
1299 2845 3 BEGIN
1300 2846 3
1301 2847 3 ! Log verify pass. Only do this for /PHYSICAL; otherwise there
1302 2848 3 ! would be a completely excessive number. Then do the verify.
1303 2849 3
1304 2850 3 SIGNAL(BACKUPS_STARTVERIFY);
1305 2851 3 COMPARE_BLOCKS(0, .INPUT_MAXBLOCK);
1306 2852 3 END
1307 2853 3 ELSE
1308 2854 3 IF NOT .OUTPUT_FILECHAR[FCH$V_NOBACKUP] OR .OUTPUT_BLOCK NEQ 1
1309 2855 3 THEN
1310 2856 3 IF .QUAL[QUAL_VOLU]
1311 2857 3 THEN
1312 2858 3 BEGIN
1313 2859 3 LOCAL
1314 2860 3 VBN: REF BBLOCK;
1315 2861 3
1316 2862 3 VBN = .INPUT_VBN_LIST[0];
1317 2863 3 WHILE .VBN NEQ INPUT_VBN_LIST[0] DO
1318 2864 3 BEGIN
1319 2865 3 COMPARE_BLOCKS(.VBN[VBN_FIRST], .VBN[VBN_LAST]);
1320 2866 3 VBN = .VBN[PLC_FLINK];
1321 2867 3 END;
1322 2868 3 END
1323 2869 3 ELSE
```

```
1324 2870 4      COMPARE_BLOCKS(1, .INPUT_MAXBLOCK);
1325 2871 3      END;
1326 2872 3
1327 2873 3
1328 2874 3      ! Write the attributes of the new file.
1329 2875 3
1330 2876 3      IF NOT .QUAL[QUAL_PHYS] AND NOT .QUAL[QUAL_COMP] AND NOT .QUAL[QUAL_IMAG]
1331 2877 3      THEN
1332 2878 4          BEGIN
1333 2879 4              LOCAL
1334 2880 4                  OLD_RECATTR:      BBLOCK[FAT$C_LENGTH],      ! Existing attributes
1335 2881 4                  FAT_ATR_LIST:      BBLOCK[12],                ! ACP attribute list
1336 2882 4                  ATR_LIST:          VECTOR[MAX_ATTRS*2 + 1]; ! New attribute list
1337 2883 4
1338 2884 4
1339 2885 4              ! Build the attribute descriptor list.
1340 2886 4              GET_ACP_ATTRIBUTES(ATR_LIST);
1341 2887 4
1342 2888 4
1343 2889 4
1344 2890 4              ! Read the record attributes. This is required because HIBLK may be
1345 2891 4              ! incorrect in the save set record attributes due to cluster factor
1346 2892 4              ! difference in the old and new media.
1347 2893 4
1348 2894 4              FAT_ATR_LIST[0,0,16,0] = ATR$S_RECATTR;
1349 2895 4              FAT_ATR_LIST[2,0,16,0] = ATR$C_RECATTR;
1350 2896 4              FAT_ATR_LIST[4,0,32,0] = OLD_RECATTR;
1351 2897 4              FAT_ATR_LIST[8,0,32,0] = 0;
1352 2898 4              STATUS = $QIOW(
1353 2899 4                  FUNC=IOS_ACCESS,
1354 2900 4                  CHAN=.OUTPUT_CHAN,
1355 2901 4                  IOSB=IOSB,
1356 2902 4                  PS=FAT_ATR_LIST);
1357 2903 4              IF .STATUS THEN STATUS = .IOSB[0];
1358 2904 4              IF NOT .STATUS
1359 2905 4              THEN
1360 2906 4                  FILE_ERROR(
1361 2907 4                      BACKUP$_READATTR,
1362 2908 4                      .OUTPUT_FAB,
1363 2909 4                      .STATUS);
1364 2910 4
1365 2911 4
1366 2912 4              ! Copy HIBLK to the new record attributes.
1367 2913 4              OUTPUT_RECATTR[FAT$L_HIBLK] = .OLD_RECATTR[FAT$L_HIBLK];
1368 2914 4
1369 2915 4
1370 2916 4
1371 2917 4              ! Write the record attributes.
1372 2918 4
1373 2919 4              STATUS = $QIOW(
1374 2920 4                  FUNC=IOS_DEACCESS,
1375 2921 4                  CHAN=.OUTPUT_CHAN,
1376 2922 4                  IOSB=IOSB,
1377 2923 4                  PS=ATR_LIST);
1378 2924 4              IF .STATUS THEN STATUS = .IOSB[0];
1379 2925 4              IF NOT .STATUS
1380 2926 4              THEN
```

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

L 7
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 46
(9)

```

1381 2927 4          FILE ERROR(
1382 2928 4          BACKUP$ WRITEATTR,
1383 2929 4          .OUTPUT_FAB,
1384 2930 4          .STATUS);
1385 2931 3          END;
1386 2932 3
1387 2933 3
1388 2934 3      ! Depending on the operation, do nothing, issue a deaccess, or issue a
1389 2935 3      deassign channel.
1390 2936 3
1391 2937 3      IF .COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]
1392 2938 3      THEN
1393 2939 4          BEGIN
1394 2940 4              IF .QUAL[QUAL_PHYS]
1395 2941 4              THEN
1396 2942 4                  STATUS = TRUE
1397 2943 4              ELSE
1398 2944 5                  BEGIN
1399 2945 5                      STATUS = $QIOW(
1400 2946 5                      FUNC=IOS$ DEACCESS,
1401 2947 5                      CHAN=.OUTPUT_CHAN,
1402 2948 5                      IOSB=IOSB);
1403 2949 5                      IF .STATUS THEN STATUS = .IOSB[0];
1404 2950 5                      END
1405 2951 4                  END
1406 2952 3              ELSE IF .QUAL[QUAL_IMAG]
1407 2953 3              THEN
1408 2954 4                  BEGIN
1409 2955 4                      STATUS = C$QIOW(
1410 2956 4                      FUNC=IOS$ DEACCESS,
1411 2957 4                      CHAN=.OUTPUT_CHAN,
1412 2958 4                      IOSB=IOSB);
1413 2959 4                      IF .STATUS THEN STATUS = .IOSB[0];
1414 2960 4                      END
1415 2961 3              ELSE
1416 2962 4                  BEGIN
1417 2963 4                      STATUS = $DASSGN(CHAN=.OUTPUT_CHAN);
1418 2964 4                      OUTPUT_CHAN = 0;
1419 2965 4                      END;
1420 2966 3
1421 2967 3
1422 2968 3      ! If the close failed, issue an appropriate message. Also log the
1423 2969 3      operation if requested.
1424 2970 3
1425 2971 3      IF .QUAL[QUAL_COMP]
1426 2972 3      THEN
1427 2973 4          BEGIN
1428 2974 4              IF NOT .STATUS
1429 2975 4              THEN
1430 2976 4                  FILE ERROR(
1431 2977 4                  BACKUP$ CLOSEIN + STS$K_ERROR,
1432 2978 4                  .OUTPUT_FAB,
1433 2979 4                  .STATUS);
1434 2980 4              IF .QUAL[QUAL_LOG]
1435 2981 4              THEN
1436 2982 4                  FILE ERROR(
1437 2983 4                  BACKUP$ COMPARED, .OUTPUT_FAB);
```


RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

M 7
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 47
(9)

```

: 1438      2984  4      END
: 1439      2985  3      ELSE
: 1440      2986  4      BEGIN
: 1441      2987  4      IF NOT .STATUS
: 1442      2988  4      THEN
: 1443      2989  4      FILE_ERROR(
: 1444      2990  4      BACKUP$_CLOSEOUT + STS$K_ERROR,
: 1445      2991  4      .OUTPUT_FAB,
: 1446      2992  4      .STATUS);
: 1447      2993  4      IF .QUAL[QUAL_LOG]
: 1448      2994  4      THEN
: 1449      2995  4      FILE_ERROR(
: 1450      2996  5      TIF .QUAL[QUAL_IF11] AND .QUAL[QUAL_OF11] AND .QUAL[QUAL_VER1]
: 1451      2997  5      THEN BACKUP$ COMPARED ! Log the verification
: 1452      2998  4      ELSE BACKUP$(CREATED),
: 1453      2999  4      .OUTPUT_FAB);
: 1454      3000  3      END;
: 1455      3001  3
: 1456      3002  3
: 1457      3003  3      ! Note that no output file is open.
: 1458      3004  3      !
: 1459      3005  3      OUTPUT_FLAGS[OUTPUT_OPEN] = FALSE;
: 1460      3006  3      OUTPUT_FLAGS[OUTPUT_IMPLICIT] = FALSE;
: 1461      3007  2      END;
: 1462      3008  1      END;
```

```

0014 0008 0081F
00000000' 00820 P.AAD: .BLKB 1
00000000 00824 .WORD 8, 20
00000000 00828 .ADDRESS COM_O_BSRDATE
00000000 00828 .LONG 0
```

```

                                OFFC 00000
                                .ENTRY CLOSE RESTORE, Save R2,R3,R4,R5,R6,R7,R8,- ; 2700
                                R9,R10,R11
                                SYSSDASSGN, R11
                                MOVAB COMPARE_BLOCKS, R10
                                MOVAB INCR_RSS, R9
                                MOVAB SYSSDIOW, R8
                                MOVAB FILE_ERROR, R7
                                MOVAB QUALF12, R6
                                MOVAB -168(SP), SP
                                BBC #3, OUTPUT_FLAGS, 5$
                                TSTL INCR_RSS
                                BEQL 2$
                                CALLS #0, INCREMENTAL_DELETE
                                CALLS #0, INCREMENTAL_SCAN
                                BRB 1$
                                MOVCS #0, (SP), #0, #64, FIB
                                MOVL #2097408, FIB
                                MOVL INCR_FIB+10, FIB+4
                                MOVW INCR_FIB+14, FIB+8
                                MOVZBL #64, FIB_DESC
                                MOVAB FIB, FIB_DESC+4

0040 8F      00      7A      0154      C6      00      9E      00002
                                5A      FA4B      CF      9E      00009
                                59      00000000' EF      9E      0000E
                                58      00000000G 00      9E      00015
                                57      00000000G 00      9E      0001C
                                56      00000000' EF      9E      00023
                                5E      FF58      CE      9E      0002A
                                68      AE      00200100 8F      D0      0004E
                                6C      AE      FF72      C9      D0      00056
                                70      AE      FF76      C9      80      0005C
                                60      AE      40      8F      9A      00062
                                64      AE      68      AE      9E      00067

                                1$:
                                69      D5      00035
                                0C      13      00037
                                00      FB      00039
                                00      FB      0003E
                                F0      11      00043
                                00      2C      00045
                                AE      0004C
                                2$:
                                8F      D0      0004E
                                C9      D0      00056
                                C9      80      0005C
                                8F      9A      00062
                                AE      9E      00067

                                2728
                                2741
                                2743
                                2744
                                2741
                                2750
                                2751
                                2752
                                2754
                                2755
                                2756
```

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

N 7

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 48
(9)

			83	7E	D4	0006C	CLRL	-(SP)	: 2765
				AF	9F	0006E	PUSHAB	P.AAD	:
				7E	7C	00071	CLRQ	-(SP)	:
				7E	D4	00073	CLRL	-(SP)	:
			74	AE	9F	00075	PUSHAB	FIB_DESC	:
				7E	7C	00078	CLRQ	-(SP)	:
			78	AE	9F	0007A	PUSHAB	IOSB	:
				36	DD	0007D	PUSHL	#54	:
			0150	C6	DD	0007F	PUSHL	OUTPUT_CHAN	:
				7E	D4	00083	CLRL	-(SP)	:
		68		0C	FB	00085	CALLS	#12, SYSSQIOW	:
		07		50	E9	00088	BLBC	STATUS, 3\$	: 2766
		50	58	AE	3C	0008B	MOVZWL	IOSB, STATUS	:
		0F		50	E8	0008F	BLBS	STATUS, 4\$	: 2767
				50	DD	00092	PUSHL	STATUS	: 2772
			0158	C6	DD	00094	PUSHL	OUTPUT_FAB	: 2771
			00000000G	8F	DD	00098	PUSHL	#BACKUP\$ WRITEATTR	: 2769
		67		03	FB	0009E	CALLS	#3, FILE_ERROR	:
	0154	C6		08	8A	000A1	BICB2	#8, OUTPUT_FLAGS	: 2775
			0150	C6	DD	000A6	PUSHL	OUTPUT_CHAN	: 2776
		6B		01	FB	000AA	CALLS	#1, SYSSDASSGN	:
				13	11	000AD	BRB	6\$	: 2777
			0154	C6	95	000AF	TSTB	OUTPUT_FLAGS	: 2779
				11	18	000B3	BGEQ	7\$	:
			0150	C6	DD	000B5	PUSHL	OUTPUT_CHAN	: 2782
		6B		01	FB	000B9	CALLS	#1, SYSSDASSGN	:
	0154	C6	80	8F	8A	000BC	BICB2	#128, OUTPUT_FLAGS	: 2783
			0150	C6	D4	000C2	CLRL	OUTPUT_CHAN	: 2784
		01	0154	C6	E8	000C6	BLBS	OUTPUT_FLAGS, 8\$	: 2787
					04	000CB	RET		:
		52	FEC0	D6	0F	000CC	REMQUE	@OUTPUT_WAIT, BCB	: 2798
				14	1D	000D1	BVS	9\$	:
				52	DD	000D3	PUSHL	BCB	: 2800
	00000000G	00		01	FB	000D5	CALLS	#1, WAIT	:
				52	DD	000DC	PUSHL	BCB	: 2801
	00000000G	00		01	FB	000DE	CALLS	#1, FREE_BUFFER	:
				E5	11	000E5	BRB	8\$	: 2798
		33	02	A6	E8	000E7	BLBS	QUAL+14, 10\$	: 2814
2D	0154	C6		02	E0	000EB	BBS	#2, OUTPUT_FLAGS, 10\$	: 2815
27	0154	C6		04	E0	000F1	BBS	#4, OUTPUT_FLAGS, 10\$	: 2816
		22	0155	C6	E8	000F7	BLBS	OUTPUT_FLAGS+1, 10\$	: 2817
	0170	C6	016C	C6	D1	000FC	CMPL	OUTPUT_BLOCK, OUTPUT_MAXBLOCK	: 2818
				19	1A	00103	BGTRU	10\$	:
		01	016C	C6	D1	00105	CMPL	OUTPUT_BLOCK, #1	: 2819
				12	13	0010A	BEQL	10\$	:
		7E	016C	C6	7D	0010C	MOVQ	OUTPUT_BLOCK, -(SP)	: 2824
			0158	C6	DD	00111	PUSHL	OUTPUT_FAB	: 2823
			00000000G	8F	DD	00115	PUSHL	#BACKUP\$ VBNMISSING	: 2821
		67		04	FB	0011B	CALLS	#4, FILE_ERROR	:
6A	03	A6		04	E1	0011E	BBC	#4, QUAL+15, 17\$	: 2829
65	03	A6		06	E1	00123	BBC	#6, QUAL+15, 17\$	:
			01	A6	95	00128	TSTB	QUAL+13	:
				60	18	0012B	BGEQ	17\$	:
0D	FF	A6		01	E1	0012D	BBC	#1, QUAL+11, 11\$	: 2835
			0158	C6	DD	00132	PUSHL	OUTPUT_FAB	: 2838
			00000000G	8F	DD	00136	PUSHL	#BACKUP\$ CREATED	: 2837
		67		02	FB	0013C	CALLS	#2, FILE_ERROR	:

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

B 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 49
(9)

15	66	00000000G	05	E1	0013F	11\$:	BBC	#5, QUAL+12, 12\$	2843	
	00000000G	00	8F	DD	00143		PUSHL	#BACKUP\$ STARTVERIFY	2850	
		00D8	01	FB	00149		CALLS	#1, LIB\$SIGNAL		
			C6	DD	00150		PUSHL	INPUT_MAXBLOCK	2851	
			7E	D4	00154		CLRL	-(SP)		
			32	11	00156		BRB	16\$		
07	01E0	C6	01	E1	00158	12\$:	BBC	#1, OUTPUT_FILECHAR, 13\$	2854	
		016C	C6	D1	0015E		CMPL	OUTPUT_BLOCK, #1		
			28	13	00163		BEQL	17\$		
		1B	02	A6	E9	00165	13\$:	BLBC	QUAL+14, 15\$	2856
		52	0144	C6	D0	00169		MOVL	INPUT_VBN_LIST, VBN	2862
		50	0144	C6	9E	0016E	14\$:	MOVAB	INPUT_VBN_LIST, R0	2863
		50		52	D1	00173		CMPL	VBN, R0	
				15	13	00176		BEQL	17\$	
		7E	0A	A2	7D	00178		MOVQ	10(VBN), -(SP)	2865
		6A		02	FB	0017C		CALLS	#2, COMPARE_BLOCKS	
		52		62	D0	0017F		MOVL	(VBN), VBN	2866
				EA	11	00182		BRB	14\$	2863
		00D8	C6	DD	00184	15\$:	PUSHL	INPUT_MAXBLOCK	2870	
			01	DD	00188		PUSHL	#1		
		6A		02	FB	0018A	16\$:	CALLS	#2, COMPARE_BLOCKS	
03		66		05	E1	0018D	17\$:	BBC	#5, QUAL+12, 19\$	2876
			0091	31	00191	18\$:	BRW	23\$		
		FC	A6	95	00194	19\$:	TSTB	QUAL+8		
			F8	19	00197		BLSS	18\$		
F3	FE	A6		03	E0	00199		BBS	#3, QUAL+10, 18\$	
			5E	DD	0019E		PUSHL	SP	2887	
	03EF	CA	01	FB	001A0		CALLS	#1, GET_ACP_ATTRIBUTES		
	74	AE	8F	D0	001A5		MOVL	#262176, FAT_ATR_LIST	2894	
	78	AE	D8	AD	9E	001AD		MOVAB	OLD_RECATTR, FAT_ATR_LIST+4	2896
			7C	AE	D4	001B2		CLRL	FAT_ATR_LIST+8	2897
			7E	D4	001B5		CLRL	-(SP)	2902	
		78	AE	9F	001B7		PUSHAB	FAT_ATR_LIST		
			7E	7C	001BA		CLRQ	-(SP)		
			7E	7C	001BC		CLRQ	-(SP)		
			7E	7C	001BE		CLRQ	-(SP)		
		F8	AD	9F	001C0		PUSHAB	IOSB		
			32	DD	001C3		PUSHL	#50		
		0150	C6	DD	001C5		PUSHL	OUTPUT_CHAN		
			7E	D4	001C9		CLRL	-(SP)		
		68	0C	FB	001CB		CALLS	#12, SYSSQIOW		
		52	50	D0	001CE		MOVL	R0, STATUS		
		07	52	E9	001D1		BLBC	STATUS, 20\$	2903	
		52	F8	AD	3C	001D4		MOVZWL	IOSB, STATUS	
		0F	52	E8	001D8		PLBS	STATUS, 21\$	2904	
			52	DD	001DB	20\$:	PUSHL	STATUS	2909	
		0158	C6	DD	001DD		PUSHL	OUTPUT_FAB	2908	
		00000000G	8F	DD	001E1		PUSHL	#BACKUP\$ READATTR	2906	
			03	FB	001E7		CALLS	#3, FILE_ERROR		
0190	67		AD	D0	001EA	21\$:	MOVL	OLD_RECATTR+4, OUTPUT_RECATTR+4	2914	
	C6	DC	7E	D4	001F0		CLRL	-(SP)	2923	
			04	AE	9F	001F2		PUSHAB	ATR_LIST	
			7E	7C	001F5		CLRQ	-(SP)		
			7E	7C	001F7		CLRQ	-(SP)		
			7E	7C	001F9		CLRQ	-(SP)		
		F8	AD	9F	001FB		PUSHAB	IOSB		
			34	DD	001FE		PUSHL	#52		

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

C 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 50
(9)

			0150	C6	DD	00200	PUSHL	OUTPUT_CHAN		
				7E	D4	00204	CLRL	-(SP)		
		68		0C	FB	00206	CALLS	#12, SYSSQIOW		
		52		50	D0	00209	MOVL	R0, STATUS		
		07		52	E9	0020C	BLBC	STATUS, 22\$		2924
		52	F8	AD	3C	0020F	MOVZWL	IOSB, STATUS		
		0F		52	E8	00213	BLBS	STATUS, 23\$		2925
				52	DD	00216	PUSHL	STATUS		2930
			0158	C6	DD	00218	PUSHL	OUTPUT_FAB		2929
			00000000G	8F	DD	0021C	PUSHL	#BACKUP\$ WRITEATTR		2927
		67		03	FB	00222	CALLS	#3, FILE_ERROR		
21	6E	A6		03	E1	00225	BBC	#3, COM_FLAGS, 25\$		2937
				03	A6	95 0022A	TSTB	QUAL+15-		
				1C	18	0022D	BGEQ	25\$		
05		66		05	E1	0022F	BBC	#5, QUAL+12, 24\$		2940
		52		01	D0	00233	MOVL	#1, STATUS		2942
				6C	11	00236	BRB	30\$		
				7E	7C	00238	CLRQ	-(SP)		2948
				7E	7C	0023A	CLRQ	-(SP)		
				7E	7C	0023C	CLRQ	-(SP)		
				7E	7C	0023E	CLRQ	-(SP)		
			F8	AD	9F	00240	PUSHAB	IOSB		
				34	DD	00243	PUSHL	#52		
			0150	C6	DD	00245	PUSHL	OUTPUT_CHAN		
				3C	11	00249	BRB	27\$		
		50	0150	C6	D0	0024B	MOVL	OUTPUT_CHAN, R0		2958
43	FE	A6		03	E1	00250	BBC	#3, QUAL+10, 29\$		2952
	0001FFFF	8F		50	D1	00255	CMPL	R0, #131071		2958
				1A	1F	0025C	BLSSU	26\$		
				7E	7C	0025E	CLRQ	-(SP)		
				7E	7C	00260	CLRQ	-(SP)		
				7E	7C	00262	CLRQ	-(SP)		
				7E	7C	00264	CLRQ	-(SP)		
			F8	AD	9F	00266	PUSHAB	IOSB		
				34	DD	00269	PUSHL	#52		
				50	DD	0026B	PUSHL	R0		
				7E	D4	0026D	CLRL	-(SP)		
		00000000G	00	0C	FB	0C26F	CALLS	#12, STA_QIOW		
				14	11	00276	BRB	28\$		
				7E	7C	00278	CLRQ	-(SP)		
				7E	7C	0027A	CLRQ	-(SP)		
				7E	7C	0027C	CLRQ	-(SP)		
				7E	7C	0027E	CLRQ	-(SP)		
			F8	AD	9F	00280	PUSHAB	IOSB		
				34	DD	00283	PUSHL	#52		
				50	DD	00285	PUSHL	R0		
		68		7E	D4	00287	CLRL	-(SP)		
		52		0C	FB	00289	CALLS	#12, SYSSQIOW		
		12		50	D0	0028C	MOVL	R0, STATUS		2959
		52		52	E9	0028F	BLBC	STATUS, 30\$		
			F8	AD	3C	00292	MOVZWL	IOSB, STATUS		
				0C	11	00296	BRB	30\$		2952
				50	DD	00298	PUSHL	R0		2963
		68		01	FB	0029A	CALLS	#1, SYSSDASSGN		
		52		50	D0	0029D	MOVL	R0, STATUS		
			0150	C6	D4	002A0	CLRL	OUTPUT_CHAN		2964
		50		52	D2	002A4	MCOML	STATUS, R0		2974

RESTORE
V04-000

Restore a save set
CLOSE_RESTORE - close restored file

D 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 51
(9)

			FC	A6	95	002A7	TSTB	QUAL+8	:	2971
				1D	18	002AA	BGEQ	32\$	:	
		OF		50	E9	002AC	BLBC	R0, 31\$	:	2974
				52	DD	002AF	PUSHL	STATUS	:	2979
			0158	C6	DD	002B1	PUSHL	OUTPUT_FAB	:	2978
			00000000G	8F	DD	002B5	PUSHL	#BACKUPS_CLOSEIN+2	:	2977
		67		03	FB	002BB	CALLS	#3, FILE_ERROR	:	
41	FF	A6		01	E1	002BE	BBC	#1, QUAL+11, 37\$	:	2980
			0158	C6	DD	002C3	PUSHL	OUTPUT_FAB	:	2983
				2A	11	002C7	BRB	34\$	:	2982
		OF		50	E9	002C9	BLBC	R0, 33\$	:	2987
				52	DD	002CC	PUSHL	STATUS	:	2992
			0158	C6	DD	002CE	PUSHL	OUTPUT_FAB	:	2991
			00000000G	8F	DD	002D2	PUSHL	#BACKUPS_CLOSEOUT+2	:	2990
		67		03	FB	002D8	CALLS	#3, FILE_ERROR	:	
24	FF	A6		01	E1	002DB	BBC	#1, QUAL+11, 37\$	:	2993
			0158	C6	DD	002E0	PUSHL	OUTPUT_FAB	:	2999
12	03	A6		04	E1	002E4	BBC	#4, QUAL+15, 35\$	:	2996
0D	03	A6		06	E1	002E9	BBC	#6, QUAL+15, 35\$	:	
			01	A6	95	002EE	TSTB	QUAL+13	:	
				08	18	002F1	BGEQ	35\$	:	
			00000000G	8F	DD	002F3	PUSHL	#BACKUPS_COMPARED	:	
				06	11	002F9	BRB	36\$	:	
			00000000G	8F	DD	002FB	PUSHL	#BACKUPS_CREATED	:	
		67		02	FB	00301	CALLS	#2, FILE_ERROR	:	
	0154	C6		11	8A	00304	BICB2	#17, OUTPUT_FLAGS	:	3006
				04	00309		RET		:	3008

; Routine Size: 778 bytes, Routine Base: CODE + 082C

RESTORE
V04-000

Restore a save set
FIN_RESTORE - finish restore or compare operati

E 8
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 52
(10)

```

: 1464 3009 1 %SBTTL 'FIN RESTORE - finish restore or compare operation'
: 1465 3010 1 GLOBAL ROUTINE FIN_RESTORE: NOVALUE=
: 1466 3011 1
: 1467 3012 1 ++
: 1468 3013 1
: 1469 3014 1 FUNCTIONAL DESCRIPTION:
: 1470 3015 1 This routine finishes a restore or compare operation.
: 1471 3016 1
: 1472 3017 1 INPUT PARAMETERS:
: 1473 3018 1 NONE
: 1474 3019 1
: 1475 3020 1 IMPLICIT INPUTS:
: 1476 3021 1 NONE
: 1477 3022 1
: 1478 3023 1 OUTPUT PARAMETERS:
: 1479 3024 1 NONE
: 1480 3025 1
: 1481 3026 1 IMPLICIT OUTPUTS:
: 1482 3027 1 NONE
: 1483 3028 1
: 1484 3029 1 ROUTINE VALUE:
: 1485 3030 1 NONE
: 1486 3031 1
: 1487 3032 1 SIDE EFFECTS:
: 1488 3033 1 NONE
: 1489 3034 1
: 1490 3035 1 --
: 1491 3036 1
: 1492 3037 2 BEGIN
: 1493 3038 2
: 1494 3039 2 ! Close the output file if one is open.
: 1495 3040 2
: 1496 3041 2 CLOSE_RESTORE();
: 1497 3042 2
: 1498 3043 2
: 1499 3044 2 ! Free the compare buffer if one was allocated.
: 1500 3045 2
: 1501 3046 2 IF .QUAL[QUAL_COMP]
: 1502 3047 2 THEN
: 1503 3048 2 BEGIN
: 1504 3049 2 FREE_BUFFER(.COMPARE_BCB);
: 1505 3050 2 COMPARE_BCB = 0;
: 1506 3051 2 END;
: 1507 3052 2
: 1508 3053 2
: 1509 3054 2 ! If this is a save verify pass, deassign the channel.
: 1510 3055 2
: 1511 3056 2 IF .COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]
: 1512 3057 2 THEN
: 1513 3058 2 BEGIN
: 1514 3059 2 $DASSGN(CHAN=.OUTPUT_CHAN);
: 1515 3060 2 OUTPUT_CHAN = 0;
: 1516 3061 2 END;
: 1517 3062 2
: 1518 3063 2
: 1519 3064 2 ! If this is an image restore, finish the file structure on the output
: 1520 3065 2 ! volumes.
```

RESTORE
V04-000

Restore a save set
FIN_RESTORE - finish restore or compare operati

F 8
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 53
(10)

```
: 1521
: 1522
: 1523
: 1524
3066 2 !
3067 2 IF .QUAL[QUAL_IMAG] AND NOT .QUAL[QUAL_COMP]
3068 2 THEN STA_DISMOUNT_OUTPUT();
3069 1 END;
```

				0004 00000	.ENTRY	FIN RESTORE, Save R2	: 3010
			EF	9E 00002	MOVAB	QUAL+8, R2	: 3041
	FCEB	52	CF	00 FB 00009	CALLS	#0, CLOSE_RESTORE	: 3046
			62	95 0000E	TSTB	QUAL+8	: 3049
			0F	18 00010	BGEQ	1\$	: 3050
		0224	C2	DD 00012	PUSHL	COMPARE_BCB	: 3056
	00000000G	00	01	FB 00016	CALLS	#1, FREE_BUFFER	: 3059
		0224	C2	D4 0001D	CLRL	COMPARE_BCB	: 3060
14	72	A2	03	E1 00021 1\$:	BBC	#3, COM_FLAGS, 2\$	: 3067
		07	A2	95 00026	TSTB	QUAL+15	: 3068
			0F	18 00029	BGEQ	2\$	: 3069
		0154	C2	DD 0002B	PUSHL	OUTPUT_CHAN	: 3068
	00000000G	00	01	FB 0002F	CALLS	#1, SYSSDASSGN	: 3069
		0154	C2	D4 00036	CLRL	OUTPUT_CHAN	: 3068
0B	02	A2	03	E1 0003A 2\$:	BBC	#3, QUAL+10, 3\$	: 3068
			62	95 0003F	TSTB	QUAL+8	: 3069
			07	19 00041	BLSS	3\$	: 3068
	00000000G	00	00	FB 00043	CALLS	#0, STA_DISMOUNT_OUTPUT	: 3069
			04	0004A 3\$:	RET		: 3069

; Routine Size: 75 bytes, Routine Base: CODE + 0B36

```
1526 3070 1 %SBTTL 'SELECT_FILE - apply selection to file'
1527 3071 1 ROUTINE SELECT_FILE(OUTPUT_SEL_DESC)=
1528 3072 1
1529 3073 1 ++
1530 3074 1
1531 3075 1 FUNCTIONAL DESCRIPTION:
1532 3076 1 This routine evaluates selection criteria for a file retrieved from
1533 3077 1 an input save set.
1534 3078 1
1535 3079 1 INPUT PARAMETERS:
1536 3080 1 OUTPUT_SEL_DESC - Pointer to descriptor that will be initialized with
1537 3081 1 selection filespec if /SELECT qualifier matches.
1538 3082 1
1539 3083 1 IMPLICIT INPUTS:
1540 3084 1 OUTPUT_FAB - Pointer to the FAB.
1541 3085 1 OUTPUT_NAM - Pointer to the NAM block.
1542 3086 1 OUTPUT_ATTBUF - The attributes from the save set.
1543 3087 1
1544 3088 1 OUTPUT PARAMETERS:
1545 3089 1 NONE
1546 3090 1
1547 3091 1 IMPLICIT OUTPUTS:
1548 3092 1 NONE
1549 3093 1
1550 3094 1 ROUTINE VALUE:
1551 3095 1 True if the file meets the selection criteria, false otherwise.
1552 3096 1
1553 3097 1 SIDE EFFECTS:
1554 3098 1 NONE
1555 3099 1
1556 3100 1 --
1557 3101 1
1558 3102 2 BEGIN
1559 3103 2 MAP
1560 3104 2 OUTPUT_SEL_DESC:REF VECTOR; ! Pointer to descriptor
1561 3105 2 LOCAL
1562 3106 2 CMDDATE: REF VECTOR, ! Pointer to command selection date
1563 3107 2 FILDATE: REF VECTOR; ! Pointer to file selection date
1564 3108 2
1565 3109 2
1566 3110 2 IF .QUAL[QUAL_BEFO] OR .QUAL[QUAL_SINC]
1567 3111 2 THEN
1568 3112 3 BEGIN
1569 3113 3 IF .QUAL[QUAL_BACK] THEN FILDATE = OUTPUT_ATTBUF[FAR_BAKDATE] ELSE
1570 3114 3 IF .QUAL[QUAL_CREA] THEN FILDATE = OUTPUT_ATTBUF[FAR_CREDATE] ELSE
1571 3115 3 IF .QUAL[QUAL_EXPI] THEN FILDATE = OUTPUT_ATTBUF[FAR_EXPDATE] ELSE
1572 3116 3 FILDATE = OUTPUT_ATTBUF[FAR_REVDATE];
1573 3117 2 END;
1574 3118 2
1575 3119 2
1576 3120 2 ! BEFORE
1577 3121 2
1578 3122 2 IF .QUAL[QUAL_BEFO]
1579 3123 2 THEN
1580 3124 3 BEGIN
1581 3125 3 IF .QUAL[QUAL_BEFO_BACK]
1582 3126 3 THEN CMDDATE = OUTPUT_ATTBUF[FAR_BAKDATE]
```


RESTORE
V04-000

Restore a save set
SELECT_FILE - apply selection to file

H 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 55
(11)

```

1583 3127 3      ELSE CMDDATE = QUAL[QUAL_BEFO_VALUE];
1584 3128 3      IF .FILDATE[1] GTRU .CMDDATE[1] THEN RETURN FALSE;
1585 3129 3      IF .FILDATE[1] EQLU .CMDDATE[1] AND .FILDATE[0] GEQU .CMDDATE[0] THEN RETURN FALSE;
1586 3130 2      END;
1587 3131 2
1588 3132 2
1589 3133 2      ! EXCLUDE
1590 3134 2
1591 3135 2      IF .QUAL[QUAL_EXCL]
1592 3136 2      THEN
1593 3137 3      BEGIN
1594 3138 3      LOCAL
1595 3139 3      EXCL:          REF BBLOCK;
1596 3140 3
1597 3141 3      EXCL = .QUAL[QUAL_EXCL_LIST];
1598 3142 3      WHILE .EXCL NEQ 0 DO
1599 3143 4      BEGIN
1600 3144 4      IF MATCH(OUTPUT_FILENAME, EXCL[QUAL_EXCL_DESC]) THEN RETURN FALSE;
1601 3145 4      EXCL = .EXCL[QUAL_NEXT];
1602 3146 3      END;
1603 3147 2      END;
1604 3148 2
1605 3149 2
1606 3150 2      ! SELECT
1607 3151 2
1608 3152 2      IF .QUAL[QUAL_SELE]
1609 3153 2      THEN
1610 3154 3      BEGIN
1611 3155 3      LOCAL
1612 3156 3      SELE:          REF BBLOCK;
1613 3157 3
1614 3158 3      SELE = .QUAL[QUAL_SELE_LIST];
1615 3159 3      IF
1616 3160 4      BEGIN
1617 3161 4      WHILE .SELE NEQ 0 DO
1618 3162 5      BEGIN
1619 3163 5      IF MATCH(OUTPUT_FILENAME, SELE[QUAL_SELE_DESC])
1620 3164 5      THEN
1621 3165 6      BEGIN
1622 3166 6      OUTPUT_SEL_DESC[0] = .BBLOCK[SELE[QUAL_SELE_DESC], DSC$W_LENGTH];
1623 3167 6      OUTPUT_SEL_DESC[1] = .BBLOCK[SELE[QUAL_SELE_DESC], DSC$A_POINTER];
1624 3168 6      EXITLOOP FALSE;
1625 3169 5      END;
1626 3170 5      SELE = .SELE[QUAL_NEXT];
1627 3171 5      END
1628 3172 4      END
1629 3173 3      THEN
1630 3174 3      RETURN FALSE;
1631 3175 2      END;
1632 3176 2
1633 3177 2
1634 3178 2      ! OWNER
1635 3179 2
1636 3180 2      IF .QUAL[QUAL_I_OWNE]
1637 3181 2      THEN
1638 3182 3      IF NOT (
1639 3183 3      (.QUAL[QUAL_I_OWN_WGRP] OR .OUTPUT_ATTBUF[FAR_UI]GROUP) EQL .QUAL[QUAL_I_OWN_GRP]) AND
```

RESTORE
V04-000

Restore a save set
SELECT_FILE - apply selection to file

I 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 56
(11)

```

1640 3184 3      (.QUAL[QUAL_I_OWN_WMEM] OR .OUTPUT_ATTBUF[FAR_UICMEMBER] EQL .QUAL[QUAL_I_OWN_MEM]))
1641 3185 2      THEN
1642 3186 2      RETURN FALSE;
1643 3187 2
1644 3188 2
1645 3189 2      ! SINCE
1646 3190 2      !
1647 3191 2      IF .QUAL[QUAL_SINC]
1648 3192 2      THEN
1649 3193 3      BEGIN
1650 3194 3      IF .QUAL[QUAL_SINC_BACK]
1651 3195 3      THEN CMDDATE = OUTPUT_ATTBUF[FAR_BAKDATE]
1652 3196 3      ELSE CMDDATE = QUAL[QUAL_SINC_VALUE];
1653 3197 3      IF .FILDATE[1] LSSU .CMDDATE[1] THEN RETURN FALSE;
1654 3198 3      IF .FILDATE[1] EQLU .CMDDATE[1] AND .FILDATE[0] LSSU .CMDDATE[0] THEN RETURN FALSE;
1655 3199 2      END;
1656 3200 2
1657 3201 2      ! VOLUME
1658 3202 2      !
1659 3203 2      !
1660 3204 2
1661 3205 2
1662 3206 2      ! Passes all tests, return true.
1663 3207 2      !
1664 3208 2      TRUE
1665 3209 1      END;
```

00FC 00000 SELECT_FILE:									
		57	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	3071
		56	00000000'	EF	9E	00009	MOVAB	MATCH, R7	
05		66		02	E0	00010	MOVAB	QUAL+8, R6	
28	05	A6		04	E1	00014	BBS	#2, QUAL+8, 1\$	3110
07		66		01	E1	00019	BBC	#4, QUAL+13, 5\$	
		54	01C8	06	9E	0001D	BBC	#1, QUAL+8, 2\$	3113
				1D	11	00022	MOVAB	OUTPUT_ATTBUF+72, FILDATE	
07	01	A6		02	E1	00024	BRB	5\$	
		54	01B0	06	9E	00029	BBC	#2, QUAL+9, 3\$	3114
				11	11	0002E	MOVAB	OUTPUT_ATTBUF+48, FILDATE	
07	01	A6		05	E1	00030	BRB	5\$	
		54	01C0	06	9E	00035	BBC	#5, QUAL+9, 4\$	3115
				05	11	0003A	MOVAB	OUTPUT_ATTBUF+64, FILDATE	
		54	01B8	06	9E	0003C	BRB	5\$	
20		66		02	E1	00041	MOVAB	OUTPUT_ATTBUF+56, FILDATE	3116
07		66		03	E1	00045	BBC	#2, QUAL+8, 9\$	3122
		55	01C8	06	9E	00049	BBC	#3, QUAL+8, 6\$	3125
				04	11	0004E	MOVAB	OUTPUT_ATTBUF+72, CMDDATE	3126
		55	08	04	9E	00050	BRB	7\$	
	04	A5	04	A4	D1	00054	MOVAB	QUAL+16, CMDDATE	3127
				07	1A	00059	CMPL	4(FILDATE), 4(CMDDATE)	3128
				08	12	0005B	BGTRU	8\$	
		65		06	D1	0005D	BNEQ	9\$	3129
				03	1F	00060	CMPL	(FILDATE), (CMDDATE)	
							BLSSU	9\$	

RESTORE
V04-000

Restore a save set
SELECT_FILE - apply selection to file

J 8
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 57
(11)

18	01	A6	008D	31	00062	8\$:	BRW	19\$		
		52	04	E1	00065	9\$:	BBC	#4, QUAL+9, 11\$		3135
			28	A6	D0	0006A	MOVL	QUAL+48, EXCL		3141
				12	13	0006E	BEQL	11\$		3142
			04	A2	9F	00070	PUSHAB	4(EXCL)		3144
			0180	C6	9F	00073	PUSHAB	OUTPUT_FILENAME		
		67		02	FB	00077	CALLS	#2, MATCH		
		75		50	E8	0007A	BLBS	R0, 19\$		
		52		62	D0	0007D	MOVL	(EXCL), EXCL		3145
				EC	11	00080	BRB	10\$		3142
29	05	A6		03	E1	00082	11\$:	BBC	#3, QUAL+13, 14\$	3152
		53	3C	A6	D0	00087	MOVL	QUAL+68, SELE		3158
				65	13	0008B	12\$:	BEQL	19\$	3161
		52	04	A3	9E	0008D	MOVAB	4(SELE), R2		3163
				52	DD	00091	PUSHL	R2		
			0180	C6	9F	00093	PUSHAB	OUTPUT_FILENAME		
		67		02	FB	00097	CALLS	#2, MATCH		
		0E		50	E9	0009A	BLBC	R0, 13\$		
		50	04	AC	D0	0009D	MOVL	OUTPUT_SEL_DESC, R0		3166
		60		62	3C	000A1	MOVZWL	(R2), (R0)		
	04	A0	04	A2	D0	000A4	MOVL	4(R2), 4(R0)		3167
				05	11	000A9	BRB	14\$		3168
		53		63	D0	000AB	13\$:	MOVL	(SELE), SELE	3170
				DB	11	000AE	BRB	12\$		3161
1A	03	A6		05	E1	000B0	14\$:	BBC	#5, QUAL+11, 16\$	3180
08	03	A6		06	E0	000B5	BBS	#6, QUAL+11, 15\$		3183
	36	A6	01E2	C6	B1	000BA	CMPW	OUTPUT_ATTBUF+98, QUAL+62		
				30	12	000C0	BNEQ	19\$		
			03	A6	95	000C2	15\$:	TSTB	QUAL+11	3184
				08	19	000C5	BLSS	16\$		
	34	A6	01E0	C6	B1	000C7	CMPW	OUTPUT_ATTBUF+96, QUAL+60		
				23	12	000CD	BNEQ	19\$		
21	05	A6		04	E1	000CF	16\$:	BBC	#4, QUAL+13, 20\$	3191
07	05	A6		05	E1	000D4	BBC	#5, QUAL+13, 17\$		3194
		55	01C8	C6	9E	000D9	MOVAB	OUTPUT_ATTBUF+72, CMDDATE		3195
				04	11	000DE	BRB	18\$		
		55	18	A6	9E	000E0	17\$:	MOVAB	QUAL+32, CMDDATE	3196
	04	A5	04	A4	D1	000E4	18\$:	CMPL	4(FILDATE), 4(CMDDATE)	3197
				07	1F	000E9	BLSSU	19\$		
				08	12	000EB	BNEQ	20\$		3198
		65		64	D1	000ED	CMPL	(FILDATE), (CMDDATE)		
				03	1E	000F0	BGEQU	20\$		
				50	D4	000F2	19\$:	CLRL	R0	
					04	000F4	RET			
		50		01	D0	000F5	20\$:	MOVL	#1, R0	3209
				04	000F8	RET				

; Routine Size: 249 bytes, Routine Base: CODE + 0B81

RESTORE
V04-000

Restore a save set
CHECK_EOF_BLOCK - check file EOF

K 8
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 58
(12)

```

: 1667 3210 1 %SBTTL 'CHECK_EOF_BLOCK - check file EOF'
: 1668 3211 1 ROUTINE CHECK_EOF_BLOCK: NOVALUE=
: 1669 3212 1
: 1670 3213 1 ++
: 1671 3214 1
: 1672 3215 1 FUNCTIONAL DESCRIPTION:
: 1673 3216 1 This routine is called when a file to be compared has been opened to
: 1674 3217 1 check the EOF position of the file against the EOF position from the
: 1675 3218 1 save set.
: 1676 3219 1
: 1677 3220 1 INPUT PARAMETERS:
: 1678 3221 1 NONE
: 1679 3222 1
: 1680 3223 1 IMPLICIT INPUTS:
: 1681 3224 1 NONE
: 1682 3225 1
: 1683 3226 1 OUTPUT PARAMETERS:
: 1684 3227 1 NONE
: 1685 3228 1
: 1686 3229 1 IMPLICIT OUTPUTS:
: 1687 3230 1 NONE
: 1688 3231 1
: 1689 3232 1 ROUTINE VALUE:
: 1690 3233 1 NONE
: 1691 3234 1
: 1692 3235 1 SIDE EFFECTS:
: 1693 3236 1 NONE
: 1694 3237 1
: 1695 3238 1 --
: 1696 3239 1
: 1697 3240 2 BEGIN
: 1698 3241 2 LOCAL
: 1699 3242 2 FAT_BUFFER: BBLOCK[FAT$C_LENGTH], ! Local buffer for attributes
: 1700 3243 2 SBK_BUFFER: BBLOCK[8], ! Local buffer for statistics block
: 1701 3244 2 STATUS, ! Status return
: 1702 3245 2 IOSB: VECTOR[4,WORD], ! I/O status block
: 1703 3246 2 ATR_LIST: BBLOCK[20]; ! Read attributes parameter
: 1704 3247 2
: 1705 3248 2
: 1706 3249 2 ! Initialize the read attributes parameter.
: 1707 3250 2
: 1708 3251 2 ATR_LIST[0,0,16,0] = ATR$S_RECATTR;
: 1709 3252 2 ATR_LIST[2,0,16,0] = ATR$C_RECATTR;
: 1710 3253 2 ATR_LIST[4,0,32,0] = FAT_BUFFER;
: 1711 3254 2 ATR_LIST[8,0,16,0] = 8;
: 1712 3255 2 ATR_LIST[10,0,16,0] = ATR$C_STATBLK;
: 1713 3256 2 ATR_LIST[12,0,32,0] = SBK_BUFFER;
: 1714 3257 2 ATR_LIST[16,0,32,0] = 0;
: 1715 3258 2
: 1716 3259 2
: 1717 3260 2 ! Read attributes of the file to be compared.
: 1718 3261 2
: 1719 P 3262 2 STATUS = ($Q)IOW(
: 1720 P 3263 2 FUNC=IOS$ ACCESS,
: 1721 P 3264 2 CHAN=.OUTPUT_CHAN,
: 1722 P 3265 2 IOSB=IOSB,
: 1723 3266 2 P5=ATR_LIST);
```

RESTORE
V04-000

Restore a save set
CHECK_EOF_BLOCK - check file EOF

L 8
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 59
(12)

```
: 1724 3267 2 IF .STATUS THEN STATUS = .IOSBL0];
: 1725 3268 2 IF NOT .STATUS
: 1726 3269 2 THEN
: 1727 3270 2 BEGIN
: 1728 3271 2 FILE_ERROR(
: 1729 3272 2 BACKUPS_READATTR,
: 1730 3273 2 .OUTPUT_FAB,
: 1731 3274 2 .STATUS);
: 1732 3275 2 RETURN;
: 1733 3276 2 END;
: 1734 3277 2
: 1735 3278 2
: 1736 3279 2 ! Check the EOF block against that determined from the file attribute record.
: 1737 3280 2 ! However, if this is the first record of a continuation tape verify pass, it
: 1738 3281 2 ! cannot be checked. In this case, the true EFBK is not available from the
: 1739 3282 2 ! file attributes. Further, if this is a restore verify pass, it cannot be
: 1740 3283 2 ! checked, because the EFBK may not yet have been written.
: 1741 3284 2
: 1742 3285 2 IF
: 1743 3286 2 (NOT .OUTPUT_FLAGS[OUTPUT_V_FIRST] OR
: 1744 3287 2 NOT .COM_FLAGS[COM_VERIFYING] OR
: 1745 3288 2 .BBLOCK[.OUTPUT_BCB[BCB_BUFFER], BBH$W_VOLNUM] LEQ 1)
: 1746 3289 2 AND
: 1747 3290 2 NOT (.COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OF11])
: 1748 3291 2 THEN
: 1749 3292 2 BEGIN
: 1750 3293 2 LOCAL
: 1751 3294 2 INPUT_EOF, ! End of file block from compare file
: 1752 3295 2 SAVE_SET_EOF; ! End of file block from save set
: 1753 3296 2
: 1754 3297 2
: 1755 3298 2 ! Compute the two EOF block values.
: 1756 3299 2
: 1757 3300 2 INPUT_EOF = ROT(.FAT_BUFFER[FAT$L_EFBK], 16);
: 1758 3301 2 IF .FAT_BUFFER[FAT$W_FFBYTE] EQL 0
: 1759 3302 2 THEN INPUT_EOF = INPUT_EOF - 1;
: 1760 3303 2 SAVE_SET_EOF = ROT(.OUTPUT_RECATTR[FAT$L_EFBK], 16);
: 1761 3304 2 IF .OUTPUT_RECATTR[FAT$W_FFBYTE] EQL 0
: 1762 3305 2 THEN SAVE_SET_EOF = SAVE_SET_EOF - 1;
: 1763 3306 2
: 1764 3307 2
: 1765 3308 2 IF .INPUT_EOF NEQ .SAVE_SET_EOF
: 1766 3309 2 THEN
: 1767 3310 2 FILE_ERROR(
: 1768 3311 2 BACKUPS_EOFMISMATCH,
: 1769 3312 2 .OUTPUT_FAB);
: 1770 3313 2 END;
: 1771 3314 2
: 1772 3315 2
: 1773 3316 2 ! Limit the highest block to be compared by the allocated size of the compare
: 1774 3317 2 ! file.
: 1775 3318 2
: 1776 3319 2 OUTPUT_MAXBLOCK = MINU(.OUTPUT_MAXBLOCK, ROT(.SBK_BUFFER[SBK$L_FILESIZE], 16));
: 1777 3320 1 END;
```

```
000C 00000 CHECK_EOF_BLOCK:
      53 00000000G 00 9E 00002 .WORD Save R2,R3 ; 3211
      52 00000000' EF 9E 00009 MOVAB FILE_ERROR, R3 ;
      5E 00040020 AE 9E 00010 MOVAB OUTPUT_FAB, R2 ;
      04 AE 00090008 8F 3D 00014 PUSHL #262176 ; 3251
      08 AE 00090008 8F D0 0001A MOVAB FAT_BUFFER, ATR_LIST+4 ; 3253
      0C AE 00090008 8F D0 0001F MOVL #589832, ATR_LIST+8 ; 3254
      50 1C AE 9E 00027 MOVAB SBK_BUFFER, ATR_LIST+12 ; 3256
      8F 10 AE D4 0002C CLRL ATR_LIST+16 ; 3257
      0001FFFF 50 F8 A2 D0 0002F MOVL OUTPUT_CHAN, R0 ; 3266
      8F 50 D1 00033 CMPL R0, #131071 ;
      1D 1F 0003A BLSSU 1$ ;
      7E D4 0003C CLRL -(SP) ;
      04 AE 9F 0003E PUSHAB ATR_LIST ;
      7E 7C 00041 CLRL -(SP) ;
      7E 7C 00043 CLRL -(SP) ;
      7E 7C 00045 CLRL -(SP) ;
      34 AE 9F 00047 PUSHAB IOSB ;
      32 DD 0004A PUSHL #50 ;
      50 DD 0004C PUSHL R0 ;
      7E D4 0004E CLRL -(SP) ;
      00000000G 00 0C FB 00050 CALLS #12, STA_QIOW ;
      1B 11 00057 BRB 2$ ;
      7E D4 00059 1$: CLRL -(SP) ;
      04 AE 9F 0005B PUSHAB ATR_LIST ;
      7E 7C 0005E CLRL -(SP) ;
      7E 7C 00060 CLRL -(SP) ;
      7E 7C 00062 CLRL -(SP) ;
      34 AE 9F 00064 PUSHAB IOSB ;
      32 DD 00067 PUSHL #50 ;
      50 DD 00069 PUSHL R0 ;
      7E D4 0006B CLRL -(SP) ;
      00000000G 9F 0C FB 0006D CALLS #12, @#SYSSQIOW ;
      07 50 14 AE 3C 00074 2$: BLBC STATUS, 3$ ; 3267
      0E 50 E8 0007B MOVZWL IOSB, STATUS ;
      50 DD 0007E 3$: BLBS STATUS, 4$ ; 3268
      62 DD 00080 PUSHL STATUS ; 3274
      00000000G 8F DD 00082 PUSHL OUTPUT_FAB ; 3273
      63 03 FB 00088 PUSHL #BACKUP$ READATTR ; 3271
      04 0008B CALLS #3, FILE_ERROR ;
      14 FC A2 01 E1 0008C 4$: RET ; 3270
      1A FF16 C2 03 E1 00091 BBC #1, OUTPUT_FLAGS, 5$ ; 3286
      50 08 A2 D0 00097 BBC #3, COM_FLAGS, 6$ ; 3287
      50 0C A0 D0 0009B MOVL OUTPUT_ECB, R0 ; 3288
      01 22 A0 B1 0009F MOVL 12(R0), R0 ;
      34 1A 000A3 CMPW 34(R0), #1 ;
      06 FF16 C2 03 E1 000A5 5$: BGTRU 9$ ; 3290
      28 FEAB C2 06 E0 000AB BBS #3, COM_FLAGS, 6$ ;
      51 2C AE 10 9C 000B1 6$: ROTL #6, QUAC+15, 9$ ;
      30 AE B5 000B6 ROTL #16, FAT_BUFFER+8, INPUT_EOF ; 3300
      02 12 000B9 TSTW FAT_BUFFER+12 ; 3301
      51 D7 000BB BNEQ 7$ ;
      50 3C A2 10 9C 000BD 7$: DECL INPUT_EOF ; 3302
      ROTL #16, OUTPUT_RECATTR+8, SAVE_SET_EOF ; 3303
```

RESTORE
V04-000

Restore a save set
CHECK_EOF_BLOCK - check file EOF

N 8
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 61
(12)

		40	A2	B5	000C2	TSTW	OUTPUT_RECATTR+12	:	3304	
			02	12	000C5	BNEQ	8\$	:		
			50	D7	000C7	DECL	SAVE_SET_EOF	:	3305	
	50		51	D1	000C9	8\$:	CMPL	INPUT_EOF, SAVE_SET_EOF	:	3308
			0B	13	000CC	BEQL	9\$	:		
			62	DD	000CE	PUSHL	OUTPUT_FAB	:	3312	
		00000000G	8F	DD	000D0	PUSHL	#BACKUP\$ EOFMISMATCH	:	3310	
		63	02	FB	000D6	CALLS	#2, FILE_ERROR	:		
51	20	AE	10	9C	000D9	9\$:	ROTL	#16, SBK_BUFFER+4, R1	:	3319
		50	A2	D0	000DE	MOVL	OUTPUT_MAXBLOCK, R0	:		
		51	50	D1	000E2	CMPL	R0, R1	:		
			03	1B	000E5	BLEQU	10\$	:		
		50	51	D0	000E7	MOVL	R1, R0	:		
	18	A2	50	D0	000EA	10\$:	MOVL	R0, OUTPUT_MAXBLOCK	:	
			04	000EE	RET			:	3320	

; Routine Size: 239 bytes, Routine Base: CODE + 0C7A

```
1779 3321 1 %SBTTL 'CREATE_DIRECTORY - create a directory file'
1780 3322 1 ROUTINE CREATE_DIRECTORY: NOVALUE=
1781 3323 1
1782 3324 1 ++
1783 3325 1
1784 3326 1 FUNCTIONAL DESCRIPTION:
1785 3327 1 This routine creates a directory in response to a file attribute
1786 3328 1 record describing a directory file.
1787 3329 1
1788 3330 1 INPUT PARAMETERS:
1789 3331 1 NONE
1790 3332 1
1791 3333 1 IMPLICIT INPUTS:
1792 3334 1 NONE
1793 3335 1
1794 3336 1 OUTPUT PARAMETERS:
1795 3337 1 NONE
1796 3338 1
1797 3339 1 IMPLICIT OUTPUTS:
1798 3340 1 NONE
1799 3341 1
1800 3342 1 ROUTINE VALUE:
1801 3343 1 NONE
1802 3344 1
1803 3345 1 SIDE EFFECTS:
1804 3346 1 NONE
1805 3347 1
1806 3348 1 --
1807 3349 1
1808 3350 2 BEGIN
1809 3351 2 LOCAL
1810 3352 2 DIR_PROT: WORD ! Protection for new directory
1811 3353 2 FILE_ID: BBLOCK[FIDSC_LENGTH], ! File ID of directory
1812 3354 2 BAKDATE: VECTOR[2], ! Backup date of directory
1813 3355 2 DESC: VECTOR[2], ! Descriptor for directory
1814 3356 2 OWNER: ! Owner UIC
1815 3357 2 STATUS: ! Status variable
1816 3358 2
1817 3359 2
1818 3360 2 ! Indicate that a file was seen.
1819 3361 2
1820 3362 2 COM_FLAGS[COM_FILESEEN] = TRUE;
1821 3363 2
1822 3364 2
1823 3365 2 ! Assign a channel to the output volume.
1824 3366 2
1825 3367 2 DESC[0] = .OUTPUT_NAM[NAM$B_DEV];
1826 3368 2 DESC[1] = .OUTPUT_NAM[NAM$L_DEV];
1827 3369 2 STATUS = ASSIGN_OUTPUT_CHANNEL(DESC, OUTPUT_CHAN, 0, 0);
1828 3370 2 IF NOT .STATUS
1829 3371 2 THEN
1830 3372 3 BEGIN
1831 3373 3 SIGNAL(BACKUP$_OPENOUT + STS$K_ERROR, 1, DESC, .STATUS);
1832 3374 3 RETURN;
1833 3375 2 END;
1834 3376 2
1835 3377 2
```



```
: 1836 3378 2 ! The expanded string is either of the form dev:[dir1.dir2]dir3.DIR;1, which
: 1837 3379 2 ! we must transform to dev:[dir1.dir2.dir3] for input to CREATE_DIR, or of the
: 1838 3380 2 ! form dev:[000000]dir1.DIR;1, which we transform to dev:[dir1].
: 1839 3381 2
: 1840 3382 2 VECTOR[.OUTPUT_NAM[NAM$L_TYPE], 0; ,BYTE] =
: 1841 3383 2 .VECTOR[.OUTPUT_NAM[NAM$L_NAME], -1; ,BYTE];
: 1842 3384 2 IF
: 1843 3385 2 .OUTPUT_NAM[NAM$B_DIR] EQL 8 AND
: 1844 3386 2 CH$FIND_NOT_CH(6, -.OUTPUT_NAM[NAM$L_DIR]+1, %C'0') EQL 0
: 1845 3387 2 THEN
: 1846 3388 2 BEGIN
: 1847 3389 2 CH$MOVE(
: 1848 3390 2 .OUTPUT_NAM[NAM$B_NAME]+1,
: 1849 3391 2 .OUTPUT_NAM[NAM$L_NAME],
: 1850 3392 2 .OUTPUT_NAM[NAM$L_DIR]+1);
: 1851 3393 2 DESC[0] = .OUTPUT_NAM[NAM$B_DEV] + .OUTPUT_NAM[NAM$B_NAME] + 2;
: 1852 3394 2 END
: 1853 3395 2 ELSE
: 1854 3396 2 BEGIN
: 1855 3397 2 VECTOR[.OUTPUT_NAM[NAM$L_NAME], -1; ,BYTE] = %C'. ';
: 1856 3398 2 DESC[0] = .OUTPUT_NAM[NAM$L_TYPE] - .OUTPUT_NAM[NAM$L_DEV] + 1;
: 1857 3399 2 END;
: 1858 3400 2 OUTPUT_NAM[NAM$B_ESL] = .DESC[0];
: 1859 3401 2 DESC[1] = .OUTPUT_NAM[NAM$L_DEV];
: 1860 3402 2
: 1861 3403 2
: 1862 3404 2 ! Determine the owner UIC of the created directory according to the output
: 1863 3405 2 ! qualifiers.
: 1864 3406 2
: 1865 3407 2 OWNER = .JPI UIC;
: 1866 3408 2 IF .QUAL[QUAL_O_OWN_ORIG] OR .QUAL[QUAL_INCR] THEN OWNER = .OUTPUT_ATTBUF[FAR_UIC];
: 1867 3409 2 IF .QUAL[QUAL_O_OWN_PARE] THEN OWNER = 0;
: 1868 3410 2 IF .QUAL[QUAL_O_OWN_UIC] THEN OWNER = .QUAL[QUAL_O_OWN_VALU];
: 1869 3411 2
: 1870 3412 2 ! Set up the directory protection
: 1871 3413 2
: 1872 3414 2 IF .QUAL[_UAL_INTE]
: 1873 3415 2 THEN
: 1874 3416 2 BEGIN
: 1875 3417 2 SYSS$SETDFPROT ( 0, DIR_PROT ) ;
: 1876 3418 2 DIR_PROT = .DIR_PROT OR %X'8888' ;
: 1877 3419 2 END
: 1878 3420 2 ELSE DIR_PROT = .OUTPUT_ATTBUF[FAR_FPRO] ;
: 1879 3421 2
: 1880 3422 2 ! Create the directory.
: 1881 3423 2
: 1882 3424 2 STATUS = CREATE_DIR(
: 1883 3425 2 DESC,
: 1884 3426 2 .OWNER,
: 1885 3427 2 .DIR_PROT,
: 1886 3428 2 .OUTPUT_RECATTR[FAT$W_VERSIONS],
: 1887 3429 2 FILE_ID,
: 1888 3430 2 BAKDATE);
: 1889 3431 2 IF NOT .STATUS
: 1890 3432 2 THEN
: 1891 3433 2 BEGIN
: 1892 3434 2 SIGNAL(
```

```
1893 3435 3      BACKUPS_CREDIRERR, 1, DESC,
1894 3436 3      .STATUS;
1895 3437 3      $DASSGN(CHAN=.OUTPUT_CHAN);
1896 3438 3      OUTPUT_CHAN = 0;
1897 3439 3      RETURN;
1898 3440 3      END;
1899 3441 2
1900 3442 2
1901 3443 2      ! Log the creation if requested.
1902 3444 2
1903 3445 2      IF .QUAL[QUAL_LOG] AND .STATUS EQL SS$_CREATED
1904 3446 2      THEN
1905 3447 2          SIGNAL(
1906 3448 2              BACKUPS_CREDIR, 1, DESC);
1907 3449 2
1908 3450 2
1909 3451 2      ! Check for special /INCREMENTAL processing of saved directory contents that
1910 3452 2      ! takes place if the creation date of the save set is equal to or later than
1911 3453 2      ! the backup date of the directory.
1912 3454 2
1913 3455 2      IF
1914 3456 2          .QUAL[QUAL_INCR] AND
1915 3457 2          .OUTPUT_ATTRBUF[FAR_STRUCLEVB] EQL 2 AND
1916 3458 3          (.COM_O_BSRDATE[1] GTRU .BAKDATE[1] OR
1917 3459 4          (.COM_O_BSRDATE[1] EQL .BAKDATE[1] AND
1918 3460 3              .COM_O_BSRDATE[0] GEQU .BAKDATE[0]))
1919 3461 2      THEN
1920 3462 3          BEGIN
1921 3463 3              OUTPUT_FLAGS[OUTPUT_INCR_DIR] = TRUE;
1922 3464 3              CH$FILE(0, FIB$_LENGTH, INCR_FIB);
1923 3465 3              INCR_FIB[FIB$_DID_NUM] = .FILE_ID[FID$_NUM];
1924 3466 3              INCR_FIB[FIB$_DID_SEQ] = .FILE_ID[FID$_SEQ];
1925 3467 3              INCR_FIB[FIB$_DID_RVN] = .FILE_ID[FID$_RVN];
1926 3468 3              INCREMENTAL_SCAN();
1927 3469 3          END
1928 3470 2      ELSE
1929 3471 2          OUTPUT_FLAGS[OUTPUT_DIR_OPEN] = TRUE ;
1930 3472 2
1931 3473 2      OUTPUT_NAM[NAM$_FID_NUM] = .FILE_ID[FID$_NUM];
1932 3474 2      OUTPUT_NAM[NAM$_FID_SEQ] = .FILE_ID[FID$_SEQ];
1933 3475 2      OUTPUT_NAM[NAM$_FID_RVN] = .FILE_ID[FID$_RVN];
1934 3476 2
1935 3477 1      END;
```

07FC 00000 CREATE_DIRECTORY:

	5A	00000000'	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10
	59	00000000G	00	9E	00009	MOVAB	INCR_FIB, R10
	58	00000000'	EF	9E	00010	MOVAB	LIB\$SIGNAL, R9
	5E		1C	C2	00017	MOVAB	QUAL+12, R8
6E	A8		04	88	0001A	SUBL2	#28, SP
	50	015C	C8	D0	0001E	BISB2	#4, COM_FLAGS
04	AE	39	A0	9A	00023	MOVL	OUTPUT_NAM, R0
						MOVZBL	57(R0), DESC

: 3322

: 3362
: 3367

08	AE	44	A0	D0	00028	MOVL	68(R0), DESC+4	3368
			7E	7C	0002D	CLRL	-(SP)	3369
		0150	C8	9F	0002F	PUSHAB	OUTPUT_CHAN	
		10	AE	9F	00033	PUSHAB	DESC	
00000000G	00		04	FB	00036	CALLS	#4, ASSIGN_OUTPUT_CHANNEL	
	57		50	D0	0003D	MOVL	R0, STATUS	
	11		57	E8	00040	BLBS	STATUS, 1\$	3370
			57	DD	00043	PUSHL	STATUS	3373
		08	AE	9F	00045	PUSHAB	DESC	
			01	DD	00048	PUSHL	#1	
		00000000G	8F	DD	0004A	PUSHL	#BACKUP\$ OPENOUT+2	
	69		04	FB	00050	CALLS	#4, LIB\$SIGNAL	
			04	04	00053	RET		3372
	56	015C	C8	D0	00054	MOVL	OUTPUT_NAM, R6	3382
	52	4C	A6	D0	00059	MOVL	76(R6), R2	3383
50	86	FF	A2	90	0005D	MOVB	-1(R2), 80(R6)	
	08	3A	A6	91	00062	CMPB	58(R6), #8	3385
			32	12	00066	BNEQ	3\$	
01	A0		50	A6	D0	MOVL	72(R6), R0	3386
			06	30	3B	SKPC	#48, #6, 1(R0)	
				02	12	BNEQ	2\$	
				51	D4	CLRL	R1	
				51	D5	TSTL	R1	
				21	12	BNEQ	3\$	
			51	A6	9A	MOVZBL	59(R6), R1	3390
				51	D6	INCL	R1	
01	A0		50	A6	D0	MOVL	72(R6), R0	3392
	62	48	51	28	00083	MOVCL3	R1, (R2), 1(R0)	
	50	39	A6	9A	00088	MOVZBL	57(R6), R0	3393
	51	3B	A6	9A	0008C	MOVZBL	59(R6), R1	
	50		51	C0	00090	ADDL2	R1, R0	
	04	AE	02	A0	9E	MOVAB	2(R0), DESC	
				0F	11	BRB	4\$	3384
	FF	A2		2E	90	MOVB	#46, -1(R2)	3397
50	50	A6	44	A6	C3	SUBL3	68(R6), 80(R6), R0	3398
	04	AE	01	A0	9E	MOVAB	1(R0), DESC	
	0B	A6	04	AE	90	MOVB	DESC, 11(R6)	3400
	08	AE	44	A6	D0	MOVL	68(R6), DESC+4	3401
	52	FECB	C8	D0	000B3	MOVL	JPI_UIC, OWNER	3407
05	68		02	E0	000B8	BBS	#2, QUAL+12, 5\$	3408
05	FE	A8	04	E1	000BC	BBC	#4, QUAL+10, 6\$	
	52	01DC	C8	D0	000C1	MOVL	OUTPUT_ATTBUF+96, OWNER	
02	68		03	E1	000C6	BBC	#3, QUAL+12, 7\$	3409
			52	D4	000CA	CLRL	OWNER	
04	68		04	E1	000CC	BBC	#4, QUAL+12, 8\$	3410
	52	34	A8	D0	000D0	MOVL	QUAL+64, OWNER	
12	02	A8	01	E1	000D4	BBC	#1, QUAL+14, 9\$	3414
			5E	D0	000D9	PUSHL	SP	3417
			7E	D4	000DB	CLRL	-(SP)	
00000000G	00		02	FB	000DD	CALLS	#2, SYS\$SETDFPROT	
	6E	8888	8F	A8	000E4	BISW2	#34952, DIR_PROT	3418
			05	11	000E9	BRB	10\$	3414
	6E	01EA	C8	B0	000EB	MOVW	OUTPUT_ATTBUF+110, DIR_PROT	3420
		0C	AE	9F	000F0	PUSHAB	BAKDATE	3424
		18	AE	9F	000F3	PUSHAB	FILE_ID	
	7E	01AA	C8	3C	000F6	MOVZWL	OUTPUT_RECATTR+30, -(SP)	3428
	7E	0C	AE	3C	000FB	MOVZWL	DIR_PROT, -(SP)	3427

RESTORE
V04-000

Restore a save set
CREATE_DIRECTORY - create a directory file

F 9
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 66
(13)

				52	DD	000FF	PUSHL	OWNER	3426	
		18	AE	9F	00101		PUSHAB	DESC	3424	
	00000000G	00	06	FB	00104		CALLS	#6, CREATE_DIR		
		57	50	DD	0010B		MOVL	R0, STATUS		
		20	57	E8	0010E		BLBS	STATUS, 11\$	3431	
			57	DD	00111		PUSHL	STATUS	3436	
		08	AE	9F	00113		PUSHAB	DESC	3434	
			01	DD	00116		PUSHL	#1		
		00000000G	8F	DD	00118		PUSHL	#BACKUPS CREDIRERR		
		69	04	FB	0011E		CALLS	#4, LIB\$SIGNAL		
		0150	C8	DD	00121		PUSHL	OUTPUT_CHAN	3437	
	00000000G	00	01	FB	00125		CALLS	#1, SY\$DASSGN		
		0150	C8	D4	0012C		CLRL	OUTPUT_CHAN	3438	
				04	00130		RET		3433	
17	FF	A8	01	E1	00131	11\$:	BBC	#1, QUAL+11, 12\$	3445	
	00000619	8F	57	D1	00136		CMPL	STATUS, #1561		
			0E	12	0013D		BNEQ	12\$		
		04	AE	9F	0013F		PUSHAB	DESC	3447	
			01	DD	00142		PUSHL	#1		
		00000000G	8F	DD	00144		PUSHL	#BACKUPS CREDIR		
		69	03	FB	0014A		CALLS	#3, LIB\$SIGNAL		
37	FE	A8	04	E1	0014D	12\$:	BBC	#4, QUAL+10, 14\$	3456	
		02	01E9	C8	91		CMPB	OUTPUT_ATTBUF+109, #2	3457	
			30	12	00157		BNEQ	14\$		
	10	AE	0090	C8	D1		CMPL	COM_O_BSRDATE+4, BAKDATE+4	3458	
			0A	1A	0015F		BGTRU	13\$		
			26	12	00161		BNEQ	14\$	3459	
	0C	AE	008C	C8	D1		CMPL	COM_O_BSRDATE, BAKDATE	3460	
			1E	1F	00169		BLSSU	14\$		
0040	8F	00	0154	C8	08	88	0016B	13\$:	3463	
				6E	00	2C	00170		3464	
					6A		00177			
	0A	AA	14	AE	D0	00178		MOVL	FILE_ID, INCR_FIB+10	3465
	0E	AA	18	AE	B0	0017D		MOVW	FILE_ID+4, INCR_FIB+14	3467
	F208	CF		00	FB	00182		CALLS	#0, INCREMENTAL_SCAN	3468
				06	11	00187		BRB	15\$	3455
	0154	C8	80	8F	88	00189	14\$:	BISB2	#128, OUTPUT_FLAGS	3471
		50	015C	C8	D0	0018F	15\$:	MOVL	OUTPUT_NAM, R0	3473
	24	A0	14	AE	D0	00194		MOVL	FILE_ID, 36(R0)	
	28	A0	18	AE	B0	00199		MOVW	FILE_ID+4, 40(R0)	3475
				04	0019E		RET		3477	

; Routine Size: 415 bytes, Routine Base: CODE + 0D69

; 1936 3478 1

RESTORE
V04-000

Restore a save set
IMPLICIT_DIRECTORY - create implicit directory

6 9
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 67
(14)

```
: 1938 3479 1 %SBTTL 'IMPLICIT_DIRECTORY - create implicit directory file'
: 1939 3480 1 ROUTINE IMPLICIT_DIRECTORY=
: 1940 3481 1
: 1941 3482 1 |++
: 1942 3483 1 |
: 1943 3484 1 | FUNCTIONAL DESCRIPTION:
: 1944 3485 1 | This routine creates a directory when a 'directory not found' status
: 1945 3486 1 | is returned from creating a file.
: 1946 3487 1 |
: 1947 3488 1 | INPUT PARAMETERS:
: 1948 3489 1 | NONE
: 1949 3490 1 |
: 1950 3491 1 | IMPLICIT INPUTS:
: 1951 3492 1 | NONE
: 1952 3493 1 |
: 1953 3494 1 | OUTPUT PARAMETERS:
: 1954 3495 1 | NONE
: 1955 3496 1 |
: 1956 3497 1 | IMPLICIT OUTPUTS:
: 1957 3498 1 | NONE
: 1958 3499 1 |
: 1959 3500 1 | ROUTINE VALUE:
: 1960 3501 1 | True if the directory was created, false otherwise.
: 1961 3502 1 |
: 1962 3503 1 | SIDE EFFECTS:
: 1963 3504 1 | NONE
: 1964 3505 1 |
: 1965 3506 1 | --
: 1966 3507 1 |
: 1967 3508 2 BEGIN
: 1968 3509 2 LOCAL
: 1969 3510 2 DIR_PROT: WORD ; Protection for new directory
: 1970 3511 2 BAKDATE: VECTOR[2]; Backup date of directory
: 1971 3512 2 STATUS; Status variable
: 1972 3513 2 OWNER; Owner UIC
: 1973 3514 2 DESC: VECTOR[2]; Descriptor for directory
: 1974 3515 2
: 1975 3516 2
: 1976 3517 2 ! Assign a channel to the output volume.
: 1977 3518 2
: 1978 3519 2 DESC[0] = .OUTPUT_NAME[NAM$B_DEV];
: 1979 3520 2 DESC[1] = .OUTPUT_NAME[NAM$L_DEV];
: 1980 3521 2 STATUS = ASSIGN_OUTPUT_CHANNEL(DESC, OUTPUT_CHAN, 0, 0);
: 1981 3522 2 IF NOT .STATUS
: 1982 3523 2 THEN
: 1983 3524 3 BEGIN
: 1984 3525 3 SIGNAL(BACKUP$OPENOUT + STS$K_ERROR, 1, DESC, .STATUS);
: 1985 3526 3 RETURN FALSE;
: 1986 3527 2 END;
: 1987 3528 2
: 1988 3529 2
: 1989 3530 2 ! The expanded string is of the form dev:[dir1.dir2.dir3]name.typ;var.
: 1990 3531 2 ! We must strip the components following the directory specification.
: 1991 3532 2
: 1992 3533 2 DESC[0] = .OUTPUT_NAME[NAM$L_NAME] - .OUTPUT_NAME[NAM$L_DEV];
: 1993 3534 2 DESC[1] = .OUTPUT_NAME[NAM$L_DEV];
: 1994 3535 2
```

```

1995 3536 2
1996 3537 2 ! Determine the owner UIC of the created directory according to the output
1997 3538 2 ! qualifiers.
1998 3539 2
1999 3540 2 OWNER = .JPI_UIC;
2000 3541 2
2001 3542 2 IF .QUAL[QUAL_O_OWN_ORIG] OR .QUAL[QUAL_INCR]
2002 3543 2 THEN
2003 3544 2 BEGIN
2004 3545 2 OWNER = .OUTPUT_ATTBUF[FAR_DIR_UIC];
2005 3546 2 IF .OWNER EQL 0 THEN OWNER = .OUTPUT_ATTBUF[FAR_UIC];
2006 3547 2 END;
2007 3548 2
2008 3549 2 IF .QUAL[QUAL_O_OWN_PARE]
2009 3550 2 THEN
2010 3551 2 OWNER = 0;
2011 3552 2
2012 3553 2 IF .QUAL[QUAL_O_OWN_UIC]
2013 3554 2 THEN
2014 3555 2 OWNER = .QUAL[QUAL_O_OWN_VALU];
2015 3556 2
2016 3557 2 ! Set up the directory protection
2017 3558 2
2018 3559 2 IF .QUAL[QUAL_INTE]
2019 3560 2 THEN
2020 3561 2 BEGIN
2021 3562 2 SYS$SETDFPROT ( 0, DIR_PROT ) ;
2022 3563 2 DIR_PROT = .DIR_PROT OR 'X'8888' ;
2023 3564 2 END
2024 3565 2 ELSE DIR_PROT = .OUTPUT_ATTBUF[FAR_DIR_FPRO] ;
2025 3566 2
2026 3567 2
2027 3568 2 ! Create the directory. Since this is an implicit creation, we use the
2028 3569 2 ! parent ownership, protection, and default version limit recorded in the
2029 3570 2 ! file attributes.
2030 3571 2
2031 3572 2 STATUS = CREATE_DIR(
2032 3573 2 DESC,
2033 3574 2 .OWNER,
2034 3575 2 .DIR_PROT,
2035 3576 2 .OUTPUT_ATTBUF[FAR_DIR_VERLIM],
2036 3577 2 OUTPUT_NAME[NAM$W_DID],
2037 3578 2 BAKDATE);
2038 3579 2
2039 3580 2
2040 3581 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2041 3582 2 OUTPUT_CHAN = 0;
2042 3583 2
2043 3584 2
2044 3585 2 IF NOT .STATUS
2045 3586 2 THEN
2046 3587 2 BEGIN
2047 3588 2 SIGNAL(
2048 3589 2 BACKUP$CREDIRERR, 1, DESC,
2049 3590 2 .STATUS);
2050 3591 2 RETURN FALSE;
2051 3592 2 END;
```

RESTORE
V04-000

Restore a save set
IMPLICIT_DIRECTORY - create implicit directory

1 9
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 69
(14)

```
: 2052 3593 2
: 2053 3594 2
: 2054 3595 2 ! Log the creation if requested.
: 2055 3596 2
: 2056 3597 2 IF .QUAL[QUAL_LOG] AND .STATUS EQL $$$_CREATED
: 2057 3598 2 THEN
: 2058 3599 2     SIGNAL(
: 2059 3600 2         BACKUP$_CREDIR, 1, DESC);
: 2060 3601 2
: 2061 3602 2
: 2062 3603 2 TRUE
: 2063 3604 1 END;
```

003C 00000 IMPLICIT_DIRECTORY:

			55	00000000G	00	9E	00002	WORD	Save R2,R3,R4,R5	: 3480	
			54	00000000'	EF	9E	00009	MOVAB	LIB\$SIGNAL, R5		
			5E		14	C2	00010	MOVAB	QUAL+12, R4		
			50	015C	C4	D0	00013	SUBL2	#20, SP		
	04		AE	39	A0	9A	00018	MOVL	OUTPUT_NAM, R0	: 3519	
	08		AE	44	A0	D0	0001D	MOVZBL	57(R0), DESC		
					7E	7C	00022	MOVL	68(R0), DESC+4	: 3520	
				0150	C4	9F	00024	CLRQ	-(SP)	: 3521	
				10	AE	9F	00028	PUSHAB	OUTPUT_CHAN		
					04	FB	0002B	PUSHAB	DESC		
	00000000G		00		50	D0	00032	CALLS	#4, ASSIGN_OUTPUT_CHANNEL		
			53		53	E8	00035	MOVL	R0, STATUS		
			10		53	DD	00038	BLBS	STATUS, 1\$	: 3522	
					53	DD	00038	PUSHL	STATUS	: 3525	
				08	AE	9F	0003A	PUSHAB	DESC		
					01	DD	0003D	PUSHL	#1		
				00000000G	8F	DD	0003F	PUSHL	#BACKUP\$_OPENOUT+2		
					0095	31	00045	BRW	8\$		
			50	015C	C4	D0	00048	1\$:	MOVL	OUTPUT_NAM, R0	: 3533
	04	AE	4C	A0	44	A0	C3	0004D	SUBL3	68(R0), 76(R0), DESC	
			08	AE	44	A0	D0	00054	MOVL	68(R0), DESC+4	: 3534
			52	FEC8	C4	D0	00059	MOVL	JPI_UIC, OWNER	: 3540	
	05		64		02	E0	0005E	BBS	#2, QUAL+12, 2\$	: 3542	
	0C	FE	A4		04	E1	00062	BBC	#4, QUAL+10, 3\$		
			52	01F0	C4	D0	00067	2\$:	MOVL	OUTPUT_ATTBUF+116, OWNER	: 3545
					05	12	0006C	BNEQ	3\$	: 3546	
			52	01DC	C4	D0	0006E	MOVL	OUTPUT_ATTBUF+96, OWNER		
	02		64		03	E1	00073	3\$:	BBC	#3, QUAL+12, 4\$	: 3549
					52	D4	00077	CLRL	OWNER	: 3551	
	04		64		04	E1	00079	4\$:	BBC	#4, QUAL+12, 5\$	: 3553
			52	34	A4	D0	0007D	MOVL	QUAL+64, OWNER	: 3555	
	12	02	A4		01	E1	00081	5\$:	BBC	#1, QUAL+14, 6\$	: 3559
					5E	DD	00086	PUSHL	SP	: 3562	
					7E	D4	00088	CLRL	-(SP)		
	00000000G		00		02	FB	0008A	CALLS	#2, SYS\$SETDFPROT		
			6E	8888	8F	AB	00091	BISW2	#3, 952, DIR_PROT	: 3563	
					05	11	00096	BRB	7\$	: 3559	
			6E	01F4	C4	B0	00098	6\$:	MOVW	OUTPUT_ATTBUF+120, DIR_PROT	: 3565
				0C	AE	9F	0009D	7\$:	PUSHAB	BAKDATE	: 3577

RESTORE
V04-000

Restore a save set
IMPLICIT_DIRECTORY - create implicit directory

J 9
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 70
(14)

7E	015C	C4	2A	C1	000A0	ADDL3	#42, OUTPUT NAM, -(SP)	:
		7E	C4	3C	000A6	MOVZWL	OUTPUT ATT0F+124, -(SP)	:
		7E	AE	3C	000AB	MOVZWL	DIR PROT, -(SP)	:
			52	DD	000AF	PUSHL	OWNER	:
		18	AE	9F	000B1	PUSHAB	DESC	: 3572
00000000G	00		06	FB	0C0B4	CALLS	#6, CREATE_DIR	: 3577
	53		50	D0	000BB	MOVL	R0, STATUS	:
		0150	C4	DD	000BE	PUSHL	OUTPUT_CHAN	: 3581
00000000G	00		01	FB	000C2	CALLS	#1, SYSSDASSGN	:
		0150	C4	D4	000C9	CLRL	OUTPUT_CHAN	: 3582
	13		53	E8	000CD	BLBS	STATUS, 9\$	: 3585
			53	DD	000D0	PUSHL	STATUS	: 3590
		08	AE	9F	000D2	PUSHAB	DESC	: 3588
			01	DD	000D5	PUSHL	#1	:
		00000000G	8F	DD	000D7	PUSHL	#BACKUP\$ CREDIRERR	:
	65		04	FB	000DD	CALLS	#4, LIB\$SIGNAL	:
			50	D4	000E0	CLRL	R0	: 3591
			04	000E2	RET			:
17	FF	A4	01	E1	000E3	BBC	#1, QUAL+11, 10\$	: 3597
00000619	8F		53	D1	000E8	CMPL	STATUS, #1561	:
			0E	12	000EF	BNEQ	10\$	:
		04	AE	9F	000F1	PUSHAB	DESC	: 3599
			01	DD	000F4	PUSHL	#1	:
		00000000G	8F	DD	000F6	PUSHL	#BACKUP\$ CREDIR	:
	65		03	FB	000FC	CALLS	#3, LIB\$SIGNAL	:
	50		01	D0	000FF	MOVL	#1, R0	: 3604
			04	00102	RET			:

; Routine Size: 259 bytes, Routine Base: CODE + 0F08

RESTORE
V04-000

Restore a save set
INITIALIZE_AND_SELECT - initialize FAB and sele

K 9
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 71
(15)

```

2065 3605 1 %SBTTL 'INITIALIZE_AND_SELECT - initialize FAB and select file'
2066 3606 1 ROUTINE INITIALIZE_AND_SELECT=
2067 3607 1
2068 3608 1 ++
2069 3609 1
2070 3610 1 FUNCTIONAL DESCRIPTION:
2071 3611 1 This routine initializes the FAB and executes the file selection for a
2072 3612 1 file open.
2073 3613 1
2074 3614 1 INPUT PARAMETERS:
2075 3615 1 NONE
2076 3616 1
2077 3617 1 IMPLICIT INPUTS:
2078 3618 1 NONE
2079 3619 1
2080 3620 1 OUTPUT PARAMETERS:
2081 3621 1 NONE
2082 3622 1
2083 3623 1 IMPLICIT OUTPUTS:
2084 3624 1 NONE
2085 3625 1
2086 3626 1 ROUTINE VALUE:
2087 3627 1 True if the file is selected, false otherwise.
2088 3628 1
2089 3629 1 SIDE EFFECTS:
2090 3630 1 NONE
2091 3631 1
2092 3632 1 --
2093 3633 1
2094 3634 2 BEGIN
2095 3635 2 LOCAL
2096 3636 2 OUTPUT_SEL_DESC:VECTOR[2]; ! Descriptor for selection filespec
2097 3637 2
2098 3638 2
2099 3639 2 ! Always select the file during a save verify pass. Thus all files that were
2100 3640 2 saved will be verified.
2101 3641 2
2102 3642 2 IF .COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]
2103 3643 2 THEN
2104 3644 2 RETURN TRUE;
2105 3645 2
2106 3646 2
2107 3647 2 ! If this is image mode, always select the file, except that for a compare or
2108 3648 2 verify operation, the initial files are never selected. If this is a single-
2109 3649 2 volume restore, select the file only if there is a segment on the RVN.
2110 3650 2
2111 3651 2 IF .QUAL[QUAL_IMAG]
2112 3652 2 THEN
2113 3653 3 BEGIN
2114 3654 3 LOCAL
2115 3655 3 PLC: REF BBLOCK, ! Cursor for placement data
2116 3656 3 PLC_END; ! Pointer beyond placement data
2117 3657 3
2118 3658 3
2119 3659 3 IF
2120 3660 3 .QUAL[QUAL_COMP] AND
2121 3661 3 .OUTPUT_ATTBUF[FAR_FID_NMX] EQL 0 AND
```

RESTORE
V04-000

Restore a save set

INITIALIZE_AND_SELECT - initialize FAB and sele

L 9
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 72
(15)

```
: 2122 3662 4      (.OUTPUT_ATTBUF[FAR_FID_NUM] LEQU FID$C_BADBLK OR
: 2123 3663 5      (.OUTPUT_ATTBUF[FAR_STRUCLEVB] EQL 2 AND
: 2124 3664 4      .OUTPUT_ATTBUF[FAR_FID_NUM] EQL FID$C_BADLOG))
: 2125 3665 3      THEN
: 2126 3666 3      RETURN FALSE;
: 2127 3667 3
: 2128 3668 3
: 2129 3669 3      IF
: 2130 3670 3      NOT .QUAL[QUAL_VOLU] OR
: 2131 3671 3      .OUTPUT_ATTBUF[FAR_FID_RVN] EQL .QUAL[QUAL_VOLU_VALUE]
: 2132 3672 3      THEN
: 2133 3673 3      RETURN TRUE;
: 2134 3674 3
: 2135 3675 3
: 2136 3676 3      PLC = .OUTPUT_PLACEMENT[DSC$A_POINTER];
: 2137 3677 3      PLC_END = .PLC + .OUTPUT_PLACEMENT[DSC$W_LENGTH];
: 2138 3678 3      WHILE .PLC LSSA .PLC_END DO
: 2139 3679 4      BEGIN
: 2140 3680 4      LOCAL
: 2141 3681 4      TYPE;
: 2142 3682 4
: 2143 3683 4      TYPE = .(.PLC)<0,8>;
: 2144 3684 4      PLC = .PLC + 1;
: 2145 3685 4
: 2146 3686 4      CASE .TYPE FROM BSASK_PLC_FID TO BSASK_PLC_PLLBN OF
: 2147 3687 4      SET
: 2148 3688 4
: 2149 3689 4      [BSASK_PLC_FID]:
: 2150 3690 5      BEGIN
: 2151 3691 5      IF .PLC[FID$B_RVN] EQL .QUAL[QUAL_VOLU_VALUE] THEN RETURN TRUE;
: 2152 3692 5      PLC = .PLC + BSASS_PLC_FID;
: 2153 3693 4      END;
: 2154 3694 4
: 2155 3695 4      [BSASK_PLC_COUNT]:
: 2156 3696 4      PLC = .PLC + BSASS_PLC_COUNT;
: 2157 3697 4
: 2158 3698 4      [BSASK_PLC_PLACE]:
: 2159 3699 4      PLC = .PLC + BSASS_PLC_PLACE;
: 2160 3700 4
: 2161 3701 4      [BSASK_PLC_PLLBN]:
: 2162 3702 4      PLC = .PLC + BSASS_PLC_PLLBN;
: 2163 3703 4
: 2164 3704 4      [OUTRANGE]:
: 2165 3705 5      BEGIN
: 2166 3706 5      SIGNAL(BACKUP$_INVATTVAL);
: 2167 3707 5      RETURN FALSE;
: 2168 3708 4      END;
: 2169 3709 4
: 2170 3710 4      TES;
: 2171 3711 3      END;
: 2172 3712 3      RETURN FALSE;
: 2173 3713 2      END;
: 2174 3714 2
: 2175 3715 2
: 2176 3716 2 ! This is not a save verify pass, and it is not image mode.
: 2177 3717 2 ! Never restore or compare reserved files, except for the MFD
: 2178 3718 2 ! in an incremental restore.
```

```
2179 3719 2 !
2180 3720 2 IF
2181 3721 2 .OUTPUT_ATTBUF[FAR_FID_NUM] EQL 0
2182 3722 2 AND .OUTPUT_ATTBUF[FAR_FID_NUM] NEQ 0
2183 3723 2 AND .OUTPUT_ATTBUF[FAR_FID_NUM] LEQU
2184 3724 3 (IF .OUTPUT_ATTBUF[FAR_STRUCLEVB] EQL 1 THEN 5 ELSE 9)
2185 3725 3 AND (NOT .QUAL[QUAL_INCR] OR .OUTPUT_ATTBUF[FAR_FID_NUM] NEQ FID$C_MFD)
2186 3726 2 THEN
2187 3727 2 RETURN FALSE;
2188 3728 2
2189 3729 2
2190 3730 2 ! Initialize the FAB in preparation for the $PARSE.
2191 3731 2 !
2192 3732 2 OUTPUT_FAB[FAB$V_MXV] = OUTPUT_FAB[FAB$V_CIF] = OUTPUT_FAB[FAB$V_SUP] = FALSE;
2193 3733 2 OUTPUT_FAB[FAB$B_RSL] = 0;
2194 3734 2 OUTPUT_FAB[FAB$B_ESL] = 0;
2195 3735 2 OUTPUT_FAB[FAB$V_UFO] = TRUE;
2196 3736 2 OUTPUT_FAB[FAB$V_OFP] = TRUE;
2197 3737 2 OUTPUT_FAB[FAB$B_ORG] = 0;
2198 3738 2 IF .QUAL[QUAL_INCR]
2199 3739 2 THEN
2200 3740 3 BEGIN
2201 3741 3 OUTPUT_FAB[FAB$B_DNS] = %CHARCOUNT('[*...];*');
2202 3742 3 OUTPUT_FAB[FAB$B_DNA] = UPLIT BYTE('[*...];*');
2203 3743 3 OUTPUT_FAB[FAB$V_SUP] = TRUE;
2204 3744 3 END
2205 3745 2 ELSE
2206 3746 3 BEGIN
2207 3747 3 OUTPUT_FAB[FAB$B_DNS] = %CHARCOUNT(';*');
2208 3748 3 OUTPUT_FAB[FAB$B_DNA] = UPLIT BYTE(';*');
2209 3749 3 IF .QUAL[QUAL_NEQV] THEN OUTPUT_FAB[FAB$V_MXV] = TRUE;
2210 3750 3 IF .QUAL[QUAL_OVER] THEN OUTPUT_FAB[FAB$V_CIF] = TRUE;
2211 3751 3 IF .QUAL[QUAL_REPL] THEN OUTPUT_FAB[FAB$V_SUP] = TRUE;
2212 3752 2 END;
2213 3753 2
2214 3754 2
2215 3755 2 ! Set the allocation size, and select the contiguous and contiguous-best-try
2216 3756 2 ! option according to the saved file characteristics.
2217 3757 2 !
2218 3758 2 OUTPUT_FAB[FAB$V_CTG] = OUTPUT_FAB[FAB$V_CBT] = FALSE;
2219 3759 2 IF .OUTPUT_FILECHAR[FCH$V_CONTIG] THEN OUTPUT_FAB[FAB$V_CTG] = TRUE;
2220 3760 2 IF .OUTPUT_FILECHAR[FCH$V_CONTIGB] THEN OUTPUT_FAB[FAB$V_CBT] = TRUE;
2221 3761 2 OUTPUT_FAB[FAB$B_ALQ] = .OUTPUT_ATTBUF[FAR_FILESIZE];
2222 3762 2
2223 3763 2
2224 3764 2 ! If this is a restore operation, evaluate the input file selection criteria to
2225 3765 2 ! determine if the file should be restored. If it is a copy operation, the
2226 3766 2 ! input file selection criteria have already been evaluated. In any case,
2227 3767 2 ! establish the file specification that selected the file for the use of the
2228 3768 2 ! wildcard directory routine.
2229 3769 2 !
2230 3770 2 IF .QUAL[QUAL_ISAV]
2231 3771 2 THEN
2232 3772 3 BEGIN
2233 3773 3 OUTPUT_SEL_DESC[0] = %CHARCOUNT('[*]');
2234 3774 3 OUTPUT_SEL_DESC[1] = UPLIT BYTE('[*]');
2235 3775 3 IF NOT .QUAL[QUAL_INCR] THEN IF NOT SELECT_FILE(OUTPUT_SEL_DESC) THEN RETURN FALSE;
```

```

: 2236 3776 3 END
: 2237 3777 2 ELSE
: 2238 3778 3 BEGIN
: 2239 3779 3 OUTPUT_SEL_DESC[0] = .BBLOCK[INPUT_QUAL[QUAL_EXP_DESC], DSC$W_LENGTH];
: 2240 3780 3 OUTPUT_SEL_DESC[1] = .BBLOCK[INPUT_QUAL[QUAL_EXP_DESC], DSC$A_POINTER];
: 2241 3781 2 END;
: 2242 3782 2
: 2243 3783 2
: 2244 3784 2 ! Initialize the related file name block.
: 2245 3785 2
: 2246 3786 3 BEGIN
: 2247 3787 3 LOCAL
: 2248 3788 3 NAMBLK : REF BBLOCK ;
: 2249 3789 3 RLFBLK : REF BBLOCK ;
: 2250 3790 3
: 2251 3791 3 NAMBLK = .OUTPUT_FAB[FAB$S_NAM] ;
: 2252 3792 3 RLFBLK = .NAMBLK[NAM$S_RLF] ;
: 2253 3793 3 RLFBLK[NAM$B_RSL] = .OUTPUT_FILENAME[DSC$W_LENGTH];
: 2254 3794 3 RLFBLK[NAM$S_RSA] = .OUTPUT_FILENAME[DSC$A_POINTER];
: 2255 3795 3 WILD_DIRECTORY(OUTPUT_SEL_DESC, .RLFBLK );
: 2256 3796 2 END ;
: 2257 3797 2
: 2258 3798 2
: 2259 3799 2 ! Execute the $PARSE.
: 2260 3800 2
: 2261 3801 3 IF NOT $PARSE(FAB=.OUTPUT_FAB)
: 2262 3802 2 THEN
: 2263 3803 2 IF .OUTPUT_FAB[FAB$S_STS] NEQ RMSS_DNF
: 2264 3804 2 THEN
: 2265 3805 3 BEGIN
: 2266 3806 3 FILE_ERROR(
: 2267 3807 4 (IF .QUAL[QUAL_COMP]
: 2268 3808 4 THEN BACKUP$_OPENIN + STS$K_ERROR
: 2269 3809 3 ELSE BACKUP$_OPENOUT + STS$R_ERROR),
: 2270 3810 3 .OUTPUT_FAB
: 2271 3811 3 .OUTPUT_FAB[FAB$S_STS], .OUTPUT_FAB[FAB$S_STV]);
: 2272 3812 3 RETURN FALSE;
: 2273 3813 2 END;
: 2274 3814 2
: 2275 3815 2
: 2276 3816 2 ! If this is /INCREMENTAL mode, always select the file.
: 2277 3817 2
: 2278 3818 2 IF .QUAL[QUAL_INCR] THEN RETURN TRUE;
: 2279 3819 2
: 2280 3820 2
: 2281 3821 2 ! Reject directory files that are in the save set but were not selected when
: 2282 3822 2 ! they were saved.
: 2283 3823 2
: 2284 3824 2 IF
: 2285 3825 2 .OUTPUT_ATTBUF[FAR_DIR_STATUS] NEQ 0 AND
: 2286 3826 2 NOT .BBLOCK[OUTPUT_ATTBUF[FAR_DIR_STATUS], D_STAT_DIR_SEL]
: 2287 3827 2 THEN
: 2288 3828 2 RETURN FALSE;
: 2289 3829 2
: 2290 3830 2
: 2291 3831 2 ! Evaluate the /CONFIRM qualifier, now that the expanded string is known.
: 2292 3832 2
```

```
: 2293 3833 2 IF .QUAL[QUAL_CONF] AND NOT .QUAL[QUAL_IF11]
: 2294 3834 2 THEN
: 2295 3835 2 BEGIN
: 2296 3836 2 LOCAL
: 2297 3837 2     FAO_BUFFER:    VECTOR[256,BYTE],
: 2298 3838 2     FAO_DESC:    BBLOCK[8],
: 2299 3839 2     ANS_BUFFER:  VECTOR[8,BYTE],
: 2300 3840 2     ANS_DESC:    BBLOCK[8];
: 2301 3841 2
: 2302 3842 2     FAO_DESC[DSC$W_LENGTH] = 256;
: 2303 3843 2     FAO_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 2304 3844 2     FAO_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
: 2305 3845 2     FAO_DESC[DSC$A_POINTER] = FAO_BUFFER;
: 2306 3846 2     ANS_DESC[DSC$W_LENGTH] = 8;
: 2307 3847 2     ANS_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 2308 3848 2     ANS_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
: 2309 3849 2     ANS_DESC[DSC$A_POINTER] = ANS_BUFFER;
: 2310 3850 2     $FAO(
: 2311 3851 2         (IF .QUAL[QUAL_COMP]
: 2312 3852 2             THEN $DESCRIPTOR('!AD, compare? (Y or N): ')
: 2313 3853 2             ELSE $DESCRIPTOR('!AD, copy? (Y or N): ')),
: 2314 3854 2         FAO_DESC,
: 2315 3855 2         FAO_DESC,
: 2316 3856 2         .OUTPUT_NAM[NAM$B_ESL],
: 2317 3857 2         .OUTPUT_NAM[NAM$B_ESA]);
: 2318 3858 2     LIB$GET_COMMAND(ANS_DESC, FAO_DESC);
: 2319 3859 2     IF .ANS_BUFFER[0] NEQ %C'y' AND .ANS_BUFFER[0] NEQ %C'y' THEN RETURN FALSE;
: 2320 3860 2     END;
: 2321 3861 2
: 2322 3862 2
: 2323 3863 2 ! Note that the file is selected.
: 2324 3864 2
: 2325 3865 2 TRUE
: 2326 3866 1 END;
```

```
2A 3B 5D 2E 2E 2E 2A 5B 0100B P.AAE: .ASCII \[*...];*\
2A 3B 01013 P.AAF: .ASCII \;*\
5D 2A 5B 01015 P.AAG: .ASCII \[*]\
28 20 3F 65 72 61 70 6D 6F 63 20 2C 44 41 21 01018 P.AAI: .ASCII \!AD, compare? (Y or N): \
20 3A 29 4E 20 72 6F 20 59 01027
00000018 01030 P.AAH: .LONG 24
00000000 01034 .ADDRESS P.AAI
6F 20 59 28 20 3F 79 70 6F 63 20 2C 44 41 21 01038 P.AAK: .ASCII \!AD, copy? (Y or N): \
20 3A 29 4E 20 72 01047
01040
00000015 01050 P.AAJ: .BLKB 3
00000000 01054 .LONG 21
. ADDRESS P.AAK
. EXTRN SYSSPARSE
```

003C 0000 INITIALIZE_AND_SELECT:

```
55 AE AF 9E 00002 .WORD Save R2,R3,R4,R5
54 00000000 EF 9E 00006 MOVAB P.AAE, R5
5E FEE0 CE 9E 00000 MOVAB QUAL+8, R4
MOVAB -288(SP), SP
```

3606

```

Restore a save set
INITIALIZE_AND_SELECT - initialize FAB and sele

```

Address	Op-Code	Op-Code Hex	Op-Code Dec	Op-Code Name	Op-Code Comment	Op-Code Value	Op-Code Name	Op-Code Comment	Op-Code Value
08	72	A4	07	03	E1	00012	BBC	#3, COM_FLAGS, 2\$	3642
				03	A4	95	TSTB	QUAL+15-	
				03	18	0001A	BGEQ	2\$	
				022A	31	0001C	BRW	32\$	
74	02	A4		03	E1	0001F	BBC	#3, QUAL+10, 11\$	3651
				64	95	00024	TSTB	QUAL+8	3660
				1B	18	00026	BGEQ	3\$	
			01D5	C4	95	00028	TSTB	OUTPUT_ATTBUF+85	3661
				15	12	0002C	BNEQ	3\$	
		03	01D0	C4	B1	0002E	CMPL	OUTPUT_ATTBUF+80, #3	3662
				48	1B	00033	BLEQU	6\$	
		02	01ED	C4	91	00035	CMPL	OUTPUT_ATTBUF+109, #2	3663
				07	12	0003A	BNEQ	3\$	
		09	01D0	C4	B1	0003C	CMPL	OUTPUT_ATTBUF+80, #9	3664
				3A	13	00041	BEQL	6\$	
		D5	06	A4	E9	00043	BLBC	QUAL+14, 1\$	3670
	47	A4	01D4	C4	91	00047	CMPL	OUTPUT_ATTBUF+84, QUAL+79	3671
				CF	13	0004D	BEQL	1\$	
		52	018C	C4	D0	0004F	MOVL	OUTPUT_PLACEMENT+4, PLC	3676
		53	0188	C4	3C	00054	MOVZWL	OUTPUT_PLACEMENT, PLC_END	3677
		53		52	C0	00059	ADDL2	PLC, PLC_END	
		53		52	D1	0005C	CMPL	PLC, PLC_END	3678
				1C	1E	0005F	BGEQU	6\$	
		5C		82	9A	00061	MOVZBL	(PLC)+, TYPE	3683
002B	03	01	50	CF	00064	CASEL	TYPE, #1, #3	3686	
0026		0021	001B		00068	.WORD	7\$-5\$,-		
							8\$-5\$,-		
							9\$-5\$,-		
							10\$-5\$		
							#BACKUP\$ INVATTVAL	3706	
	00000000G	00	00000000G	8F	DD	00070	PUSHL	#1, LIB\$SIGNAL	3707
				01	FB	00076	CALLS	31\$	3691
				01C6	31	0007D	BRW	4(PLC), QUAL+79	3692
	47	A4	04	A2	91	00080	CMPL	1\$	3696
				95	13	00085	BEQL	9\$	
				05	11	00087	BRB	#4, PLC	3699
		52		04	C0	00089	ADDL2	4\$	3702
				CE	11	0008C	BRB	#6, PLC	3721
		52		06	C0	0008E	ADDL2	4\$	3722
				C9	11	00091	BRB	#10, PLC	3724
		52		0A	C0	00093	ADDL2	4\$	
				C4	11	00096	BRB	OUTPUT_ATTBUF+85	
			01D5	C4	95	00098	TSTB	14\$	
				2A	12	0009C	BNEQ	OUTPUT_ATTBUF+80	
			01D0	C4	B5	0009E	TSTW	14\$	
				24	13	000A2	BEQL	OUTPUT_ATTBUF+109, #1	
		01	01ED	C4	91	000A4	CMPL	#5, R0	
				05	12	000A9	BNEQ	#9, R0	
		50		05	D0	000AB	MOVL	#0, #16, OUTPUT_ATTBUF	

	62	02000006	8F	CA	000D1	BICL2	#33554438, (R2)	
	50	0160	C4	D0	000D8	MOVL	OUTPUT_NAM, R0	3733
		03	A0	94	000DD	CLRB	3(R0)	
		0B	A0	94	000E0	CLRB	11(R0)	3734
	02	A2	2002	8F	A8	BISW2	#8194, 2(R2)	3735
		1D	A1	94	000E9	CLRB	29(R1)	3737
0A	02	A4		04	E1	BBC	#4, QUAL+10, 15\$	3738
	35	A1		08	90	MOVW	#8, 53(R1)	3741
	30	A1		65	9E	MOVAB	P.AAE, 48(R1)	3742
				1F	11	BRB	18\$	3743
	35	A1		02	90	MOVW	#2, 53(R1)	3747
	30	A1	08	A5	9E	MOVAB	P.AAF, 48(R1)	3748
03	03	A4		03	E1	BBC	#3, QUAL+11, 16\$	3749
		62		02	88	BISB2	#2, (R2)	
04	03	A4		04	E1	BBC	#4, QUAL+11, 17\$	3750
	03	A2		02	88	BISB2	#2, 3(R2)	
			04	A4	95	TSTB	QUAL+12	3751
				03	18	BGEQ	19\$	
		62		04	88	BISB2	#4, (R2)	
	02	A2		30	8A	BICB2	#48, 2(R2)	3758
			01E4	C4	95	TSTB	OUTPUT_FILECHAR	3759
				04	18	BGEQ	20\$	
	02	A2		10	88	BISB2	#16, 2(R2)	
04	01E4	C4		05	E1	BBC	#5, OUTPUT_FILECHAR, 21\$	3760
	02	A2		20	88	BISB2	#32, 2(R2)	
	10	A1	01DC	C4	D0	MOVL	OUTPUT_ATTBUF+92, 16(R1)	3761
1B	07	A4		05	E1	BBC	#5, QUAL+15, 22\$	3770
	F8	AD		03	D0	MOVL	#3, OUTPUT_SEL_DESC	3773
	FC	AD	0A	A5	9E	MOVAB	P.AAG, OUTPUT_SEL_DESC+4	3774
1E	02	A4		04	E0	BBS	#4, QUAL+10, 23\$	3775
			F8	AD	9F	PUSHAB	OUTPUT_SEL_DESC	
	F9D3	CF		01	FB	CALLS	#1, SELECT_FILE	
		13		50	E8	BLBS	R0, 23\$	
				72	11	BRB	26\$	
		50	00D0	C4	D0	MOVL	INPUT_QUAL, R0	3779
		50		08	C0	ADDL2	#8, R0	
	F8	AD		60	3C	MOVZWL	(R0), OUTPUT_SEL_DESC	
	FC	AD	04	A0	D0	MOVL	4(R0), OUTPUT_SEL_DESC+4	3780
		50	015C	C4	D0	MOVL	OUTPUT_FAB, R0	3791
		50	28	A0	D0	MOVL	40(R0), NAMBLK	
		50	10	A0	D0	MOVL	16(NAMBLK), RLFBLK	3792
	03	A0	0180	C4	90	MOVW	OUTPUT_FILENAME, 3(RLFBLK)	3793
	04	A0	0184	C4	D0	MOVL	OUTPUT_FILENAME+4, 4(RLFBLK)	3794
				50	DD	PUSHL	RLFBLK	3795
			F8	AD	9F	PUSHAB	OUTPUT_SEL_DESC	
00000000G	00			02	FB	CALLS	#2, WILD_DIRECTORY	
		015C		C4	DD	PUSHL	OUTPUT_FAB	3801
00000000G	00			01	FB	CALLS	#1, SYSPARSE	
	30			50	E8	BLBS	R0, 27\$	
	50	015C		C4	D0	MOVL	OUTPUT_FAB, R0	3803
0001C04A	8F	08		A0	D1	CMPL	8(R0), -#114762	
				21	13	BEQL	27\$	
	7E	08		A0	7D	MOVQ	8(R0), -(SP)	3811
				50	DD	PUSHL	R0	3810
				64	95	TSTB	QUAL+8	3807
				08	18	BGEQ	24\$	
		00000000G		8F	DD	PUSHL	#BACKUP\$_OPENIN+2	3808

RESTORE
V04-000

Restore a save set
INITIALIZE_AND_SELECT - initialize FAB and sele

E 10
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 78
(15)

00000000G	00	00000000G	06	11	001BE	BRB	25\$		
			8F	DD	001C0	PUSHL	#BACKUP\$_OPENOUT+2		3809
			04	FB	001C6	CALLS	#4, FILE_ERROR		3807
75	02	A4	77	11	001CD	BRB	31\$		3812
		01FB	04	E0	001CF	BBS	#4, QUAL+10, 32\$		3818
			C4	95	001D4	TSTB	OUTPUT_ATTBUF+123		3825
			06	13	001D8	BEQL	28\$		
66	01FB	C4	01	E1	001DA	BBC	#1, OUTPUT_ATTBUF+123, 31\$		3826
		65	01	A4	E9	BLBC	QUAL+9, 32\$		3833
60	07	A4	04	E0	001E4	BBS	#4, QUAL+15, 32\$		
	10	AE	8F	D0	001E9	MOVL	#17694976, FAO_DESC		3842
	14	AE	AE	9E	001F1	MOVAB	FAO_BUFFER, FAO_DESC+4		3845
		6E	8F	D0	001F6	MOVL	#17694728, ANS_DESC		3846
	04	AE	AE	9E	001FD	MOVAB	ANS_BUFFER, ANS_DESC+4		3849
		50	C4	D0	00202	MOVL	OUTPUT_NAM, R0		3857
			0C	A0	DD	PUSHL	12(R0)		
		7E	GB	A0	9A	MOVZBL	1(R0), -(SP)		
			18	AE	9F	PUSHAB	FAO_DESC		
			1C	AE	9F	PUSHAB	FAO_DESC		
			64	95	00214	TSTB	QUAL+8		
			06	18	00216	BGEQ	29\$		
		50	A5	9E	00218	MOVAB	P.AAH, R0		
			04	11	0021C	BRB	30\$		
		50	A5	9E	0021E	MOVAB	P.AAJ, R0		
			50	DD	00222	PUSHL	R0		
00000000G	00		05	FB	00224	CALLS	#5, SYSS\$FAO		
			10	AE	9F	PUSHAB	FAO_DESC		3858
			04	AE	9F	PUSHAB	ANS_DESC		
00000000G	00		02	FB	00231	CALLS	#2, LIB\$GET_COMMAND		
	59	8F	08	AE	91	CMPB	ANS_BUFFER, #89		3859
			0A	13	0023D	BEQL	32\$		
	79	8F	08	AE	91	CMPB	ANS_BUFFER, #121		
			03	13	00244	BEQL	32\$		
			50	D4	00246	CLRL	R0		
				04	00248	RET			
		50	01	D0	00249	MOVL	#1, R0		3866
			04	0024C	RET				

; Routine Size: 589 bytes, Routine Base: CODE + 1058


```
2328 3867 1 %SBTTL 'SELECT_INCREMENTAL - apply incremental selection criteria'
2329 3868 1 ROUTINE SELECT_INCREMENTAL=
2330 3869 1
2331 3870 1 ++
2332 3871 1
2333 3872 1 FUNCTIONAL DESCRIPTION:
2334 3873 1 This routine performs file selection for a /INCREMENTAL mode restore.
2335 3874 1
2336 3875 1 INPUT PARAMETERS:
2337 3876 1 NONE
2338 3877 1
2339 3878 1 IMPLICIT INPUTS:
2340 3879 1 OUTPUT_NAM - Pointer to name block, on which a $PARSE has been
2341 3880 1 executed.
2342 3881 1
2343 3882 1 OUTPUT PARAMETERS:
2344 3883 1 NONE
2345 3884 1
2346 3885 1 IMPLICIT OUTPUTS:
2347 3886 1 NONE
2348 3887 1
2349 3888 1 ROUTINE VALUE:
2350 3889 1 True if and only if the file passes the /INCREMENTAL restore tests.
2351 3890 1
2352 3891 1 SIDE EFFECTS:
2353 3892 1 NONE
2354 3893 1
2355 3894 1 --
2356 3895 1
2357 3896 2 BEGIN
2358 3897 2 LOCAL
2359 3898 2 FIB_DESC: VECTOR[2], : Descriptor for FIB
2360 3899 2 FIB: BBLOCK[FIB$C_LENGTH], : FIB
2361 3900 2 ATR_DESC: BBLOCK[12], : Attribute descriptor
2362 3901 2 BAKDATE: VOLATILE VECTOR[2], : Backup date
2363 3902 2 FNA_DESC: VECTOR[2], : Filename descriptor
2364 3903 2 STATUS, : General status variable
2365 3904 2 IOSB: VECTOR[4,WORD]; : I/O status block
2366 3905 2
2367 3906 2
2368 3907 2 : If the parent directory does not exist, the file is restored.
2369 3908 2
2370 3909 2 IF .OUTPUT_NAM[NAM$W_DID_NUM] EQL 0
2371 3910 2 THEN
2372 3911 2 RETURN TRUE;
2373 3912 2
2374 3913 2
2375 3914 2 : Assign a channel to the device.
2376 3915 2
2377 3916 2 FNA_DESC[0] = .OUTPUT_NAM[NAM$B_DEV];
2378 3917 2 FNA_DESC[1] = .OUTPUT_NAM[NAM$L_DEV];
2379 3918 2 STATUS = ASSIGN_OUTPUT_CHANNEL(FNA_DESC, OUTPUT_CHAN, 0, 0);
2380 3919 2 IF NOT .STATUS
2381 3920 2 THEN
2382 3921 2 BEGIN
2383 3922 2 SIGNAL(BACKUP$OPENOUT + ST$K_ERROR, 1, FNA_DESC, .STATUS);
2384 3923 2 RETURN FALSE;
```

```
2385 3924 2 END;
2386 3925 2
2387 3926 2
2388 3927 2 ! Access the parent directory to obtain the backup date.
2389 3928 2
2390 3929 2 FIB_DESC[0] = FIBSC_LENGTH;
2391 3930 2 FIB_DESC[1] = FIB;
2392 3931 2 CH$FILL(0, FIBSC_LENGTH, FIB);
2393 3932 2 FIB[FIB$W_FID_NUM] = .OUTPUT_NAM[NAM$W_DID_NUM];
2394 3933 2 FIB[FIB$W_FID_SEQ] = .OUTPUT_NAM[NAM$W_DID_SEQ];
2395 3934 2 FIB[FIB$W_FID_RVN] = .OUTPUT_NAM[NAM$W_DID_RVN];
2396 3935 2 ATR_DESC[0,0,16,0] = ATR$S_BAKDATE;
2397 3936 2 ATR_DESC[2,0,16,0] = ATR$C_BAKDATE;
2398 3937 2 ATR_DESC[4,0,32,0] = BAKDATE;
2399 3938 2 ATR_DESC[8,0,32,0] = 0;
2400 3939 2 STATUS = $QIOW(
P 3940 2 FUNC=IOS_ACCESS,
P 3941 2 CHAN=.OUTPUT_CHAN,
P 3942 2 IOSB=IOSB,
P 3943 2 P1=FIB_DESC,
2405 3944 2 P5=ATR_DESC);
2406 3945 2 IF .STATUS THEN STATUS = .IOSB[0];
2407 3946 2 IF NOT .STATUS
2408 3947 2 THEN
2409 3948 2 BEGIN
2410 3949 2 FILE_ERROR(BACKUP$ READATTR, .OUTPUT_FAB, .STATUS);
2411 3950 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2412 3951 2 OUTPUT_CHAN = 0;
2413 3952 2 RETURN FALSE;
2414 3953 2 END;
2415 3954 2
2416 3955 2
2417 3956 2 ! If the creation date of the save set is later than the backup date of the
2418 3957 2 ! parent directory, the file is restored.
2419 3958 2
2420 3959 2 IF
2421 3960 2 (.COM_O_BSRDATE[1] GTRU .BAKDATE[1] OR
2422 3961 2 (.COM_O_BSRDATE[1] EQL .BAKDATE[1] AND .COM_O_BSRDATE[0] GTRU .BAKDATE[0])
2423 3962 2 THEN
2424 3963 2 BEGIN
2425 3964 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2426 3965 2 OUTPUT_CHAN = 0;
2427 3966 2 RETURN TRUE;
2428 3967 2 END;
2429 3968 2
2430 3969 2
2431 3970 2 ! The creation date of the save set is equal to or earlier than the backup date
2432 3971 2 ! of the parent directory. Attempt to look up the file in the directory.
2433 3972 2
2434 3973 2 CH$FILL(0, FIBSC_LENGTH, FIB);
2435 3974 2 FIB[FIB$W_DID_NUM] = .OUTPUT_NAM[NAM$W_DID_NUM];
2436 3975 2 FIB[FIB$W_DID_SEQ] = .OUTPUT_NAM[NAM$W_DID_SEQ];
2437 3976 2 FIB[FIB$W_DID_RVN] = .OUTPUT_NAM[NAM$W_DID_RVN];
2438 3977 2 FNA_DESC[0] = .OUTPUT_NAM[NAM$B_NAME] + .OUTPUT_NAM[NAM$B_TYPE] + .OUTPUT_NAM[NAM$B_VER];
2439 3978 2 FNA_DESC[1] = .OUTPUT_NAM[NAM$B_NAME];
P 3979 2 STATUS = $QIOW(
P 3980 2 FUNC=IOS_ACCESS,
```

```
2442 P 3981 2 CHAN=.OUTPUT_CHAN,
2443 P 3982 2 IOSB=IOSB,
2444 P 3983 2 P1=FIB_DESC,
2445 3984 2 P2=FNA_DESC);
2446 3985 2 IF .STATUS THEN STATUS = .IOSB[0];
2447 3986 2
2448 3987 2
2449 3988 2 ! If no directory entry exists for the file, the file is not restored.
2450 3989 2
2451 3990 2 IF .STATUS EQL SSS_NOSUCHFILE
2452 3991 2 THEN
2453 3992 2 BEGIN
2454 3993 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2455 3994 2 OUTPUT_CHAN = 0;
2456 3995 2 RETURN FALSE;
2457 3996 2 END;
2458 3997 2
2459 3998 2
2460 3999 2 IF NOT .STATUS
2461 4000 2 THEN
2462 4001 2 BEGIN
2463 4002 2 FILE_ERROR(BACKUP$ OPENOUT + ST$K_ERROR, .OUTPUT_FAB, .STATUS);
2464 4003 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2465 4004 2 OUTPUT_CHAN = 0;
2466 4005 2 RETURN FALSE;
2467 4006 2 END;
2468 4007 2
2469 4008 2
2470 4009 2 ! If an entry pointing to an all-ones file ID exists, the file is restored.
2471 4010 2
2472 4011 2 IF
2473 4012 2 .FIB[FIB$W_FID_NUM] EQL 65535 AND
2474 4013 2 .FIB[FIB$W_FID_SEQ] EQL 65535 AND
2475 4014 2 .FIB[FIB$B_FID_NMX] EQL 255
2476 4015 2 THEN
2477 4016 2 BEGIN
2478 4017 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2479 4018 2 OUTPUT_CHAN = 0;
2480 4019 2 RETURN TRUE;
2481 4020 2 END;
2482 4021 2
2483 4022 2
2484 4023 2 ! Access the file to obtain the backup date. If this fails with
2485 4024 2 ! a NOSUCHFILE error, we have a dangling directory entry, which
2486 4025 2 ! is always selected.
2487 4026 2
2488 4027 2 FIB[FIB$W_DID_NUM] = FIB[FIB$W_DID_SEQ] = FIB[FIB$W_DID_RVN] = 0;
2489 P 4028 2 STATUS = $QIOQ(
2490 P 4029 2 FUNC=IOS ACCESS,
2491 P 4030 2 CHAN=.OUTPUT_CHAN,
2492 P 4031 2 IOSB=IOSB,
2493 P 4032 2 P1=FIB_DESC,
2494 4033 2 P5=ATR_DESC);
2495 4034 2 IF .STATUS THEN STATUS = .IOSB[0];
2496 4035 2
2497 4036 2 $DASSGN(CHAN=.OUTPUT_CHAN);
2498 4037 2 OUTPUT_CHAN = 0;
```

RESTORE
V04-000

Restore a save set
SELECT_INCREMENTAL - apply incremental selectio

I 10
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 82
(16)

```

2499 4038 2 IF .STATUS EQL $$$_NOSUCHFILE
2500 4039 2 THEN RETURN TRUE;
2501 4040 2
2502 4041 2 IF NOT .STATUS
2503 4042 2 THEN
2504 4043 2 BEGIN
2505 4044 2 FILE ERROR(BACKUP$_READATTR, .OUTPUT_FAB, .STATUS);
2506 4045 2 RETURN FALSE;
2507 4046 2 END;
2508 4047 2
2509 4048 2
2510 4049 2 ! If the file's backup date is non-zero and equal to or earlier than the
2511 4050 2 ! creation date of the save set, the file is restored. Otherwise, the file
2512 4051 2 ! is not restored.
2513 4052 2
2514 4053 2 IF
2515 4054 2 (.BAKDATE[0] OR .BAKDATE[1]) NEQ 0 AND
2516 4055 2 (.BAKDATE[1] LSSU .COM_0_BSRDATE[1] OR
2517 4056 2 .BAKDATE[1] EQL .COM_0_BSRDATE[1] AND .BAKDATE[0] LEQU .COM_0_BSRDATE[0])
2518 4057 2 THEN
2519 4058 2 TRUE
2520 4059 2 ELSE
2521 4060 2 FALSE
2522 4061 1 END;
```

OFFC 00000 SELECT_INCREMENTAL:

					.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 3868		
	5B	00000000G	8F	D0	00002	MOVL	#BACKUP\$_READATTR, R11		
	5A	00000000G	8F	D0	00009	MOVL	#BACKUP\$_OPENOUT+2, R10		
	59	00000000G	00	9E	00010	MOVAB	SYSSDASSGN, R9		
	58	00000000G	00	9E	00017	MOVAB	SYSSQIOW, R8		
	57	00000000G	EF	9E	0001E	MOVAB	OUTPUT_CHAN, R7		
	5E		94	AE	9E	00025	MOVAB	-108(SP), SP	
	50		0C	A7	D0	0C029	MOVL	OUTPUT_NAM, R0	3909
			2A	A0	B5	0002D	TSTW	42(R0)	
				03	12	00030	BNEQ	1\$	
				01AC	31	00032	BRW	16\$	
	08	AE	39	A0	9A	00035	MOVZBL	57(R0), FNA_DESC	3916
	0C	AE	44	A0	D0	0003A	MOVL	68(R0), FNA_DESC+4	3917
				7E	7C	0003F	CLRQ	-(SP)	3918
				57	DD	00041	PUSHL	R7	
			14	AE	9F	00043	PUSHAB	FNA_DESC	
	00000000G	00		04	FB	00046	CALLS	#4, -ASSIGN_OUTPUT_CHANNEL	
		56		50	D0	0004D	MOVL	R0, STATUS	
		13		56	E8	00050	BLBS	STATUS, 2\$	3919
				56	DD	00053	PUSHL	STATUS	3922
			0C	AE	9F	00055	PUSHAB	FNA_DESC	
				01	DD	00058	PUSHL	#1	
				5A	DD	0005A	PUSHL	R10	
	00000000G	00		04	FB	0005C	CALLS	#4, LIB\$SIGNAL	
				017F	31	00063	BRW	17\$	3923
	64	AE	40	8F	9A	00066	MOVZBL	#64, FIB_DESC	3929
	68	AE	24	AE	9E	0006B	MOVAB	FIB, FIB_DESC+4	3930

RESTORE
V04-000

Restore a save set
SELECT_INCREMENTAL - apply incremental selectio

J 10

16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 83
(16)

0040	8F	00	6E	00	2C	00070	MOV C5	#0, (SP), #0, #64, FIB	3931		
			24	AE		00077					
			50	A7	D0	00079	MOVL	OUTPUT_NAM, R0	3932		
			2A	A0	D0	0007D	MOVL	42(R0), FIB+4			
			2E	A0	B0	00082	MOVW	46(R0), FIB+8	3934		
			08	8F	D0	00087	MOVL	#1310728, ATR_DESC	3935		
			10	AE	9E	0008F	MOVAB	BAKDATE, ATR_DESC+4	3937		
			20	AE	D4	00094	CLRL	ATR_DESC+8	3938		
			7E	D4	00097	CLRL	-(SP)	3944			
			1C	AE	9F	00099	PUSHAB	ATR_DESC			
			7E	7C	0009C	CLRL	-(SP)				
			7E	D4	0009E	CLRL	-(SP)				
			78	AE	9F	000A0	PUSHAB	FIB_DESC			
			7E	7C	000A3	CLRL	-(SP)				
			20	AE	9F	000A5	PUSHAB	IOSB			
			32	DD	000A8	PUSHL	#50				
			67	DD	000AA	PUSHL	OUTPUT_CHAN				
			7E	D4	000AC	CLRL	-(SP)				
			68	0C	FB	000AE	CALLS	#12, SYSSQIOW			
			56	50	D0	000B1	MOVL	R0, STATUS			
			06	56	E9	000B4	BLBC	STATUS, 3\$	3945		
			56	6E	3C	000B7	MOVZWL	IOSB, STATUS			
			09	56	E8	000BA	BLBS	STATUS, 4\$	3946		
				56	DD	000BD	PUSHL	STATUS	3949		
			08	A7	DD	000BF	PUSHL	OUTPUT_FAB			
				5B	DD	000C2	PUSHL	R11			
				7F	11	000C4	BRB	8\$			
			14	AE	FF40	C7	D1	000C6	4\$:		
				0E	1A	000CC	CMPL	COM_O_BSRDATE+4, BAKDATE+4	3960		
			14	AE	FF40	C7	D1	000CE	5\$:		
				0B	12	000D4	BGTRU	5\$			
			10	AE	FF3C	C7	D1	000D6	CMPL	COM_O_BSRDATE+4, BAKDATE+4	3961
				03	1B	000DC	BNEQ	6\$			
				008B	31	000DE	CMPL	COM_O_BSRDATE, BAKDATE			
				00	2C	000E1	BLEQU	6\$	5\$:		
			6E	AE		000E8	BRW	11\$	6\$:		
			24	A7	D0	000EA	MOV C5	#0, (SP), #0, #64, FIB	3973		
			50	A0	D0	000EE	MOVL	OUTPUT_NAM, R0	3974		
			2E	A0	B0	000F3	MOVL	42(R0), FIB+10			
			32	A0	9A	000F8	MOVW	46(R0), FIB+14	3976		
			51	A0	9A	000FC	MOVZBL	59(R0), R1	3977		
			52	A0	9A	00100	MOVZBL	60(R0), R2			
			51	52	C0	00100	ADDL2	R2, R1			
			53	A0	9A	00103	MOVZBL	61(R0), R3			
			51	53	C1	00107	ADDL3	R3, R1, FNA_DESC			
			0C	A0	D0	0010C	MOVL	76(R0), FNA_DESC+4	3978		
				7E	7C	00111	CLRL	-(SP)	3984		
				7E	7C	00113	CLRL	-(SP)			
			18	AE	9F	00115	PUSHAB	FNA_DESC			
			78	AE	9F	00118	PUSHAB	FIB_DESC			
				7E	7C	0011B	CLRL	-(SP)			
			20	AE	9F	0011D	PUSHAB	IOSB			
				32	DD	00120	PUSHL	#50			
				67	DD	00122	PUSHL	OUTPUT_CHAN			
				7E	D4	00124	CLRL	-(SP)			
			68	0C	FB	00126	CALLS	#12, SYSSQIOW			
			56	50	D0	00129	MOVL	R0, STATUS			
			03	56	E9	0012C	BLBC	STATUS, 7\$	3985		

00000910	56 8F		6E 56	3C D1	0012F 00132		MOVZWL Cmpl	IOSB, STATUS STATUS, #2320		3990
	17		11 56	13 E8	00139 0013B	7\$:	BEQL BLBS	9\$ STATUS, 10\$		3999
		08	56 A7	DD DD	0013E 00140		PUSHL PUSHL	STATUS OUTPUT_FAB		4002
00000000G	00		5A 03	DD FB	00143 00145	8\$:	PUSHL CALLS	R10 #3, FILE ERROR		
	69		67 01	DD FB	0014C 0014E	9\$:	PUSHL CALLS	OUTPUT_CHAN #1, SYSSDASSGN		4003
			67 6A	D4 11	00151 00153		CLRL BRB	OUTPUT_CHAN 14\$		4004
FFFF	8F	28	AE 18	B1 12	00155 0015B	10\$:	CMPW BNEQ	FIB+4, #65535 12\$		4005
FFFF	8F	2A	AE 10	B1 12	0015D 00163		CMPW BNEQ	FIB+6, #65535 12\$		4013
FF	8F	2D	AE 09	91 12	00165 0016A		CMPB BNEQ	FIB+9, #255 12\$		4014
	69		67 01	DD FB	0016C 0016E	11\$:	PUSHL CALLS	OUTPUT_CHAN #1, SYSSDASSGN		4017
			67 6C	D4 11	00171 00173		CLRL BRB	OUTPUT_CHAN 16\$		4018
		30	AE 2E	D4 B4	00175 00178	12\$:	CLRL CLRW	FIB+12 FIB+10		4019
			7E 7E	D4 7C	0017B 00180		CLRL PUSHAB	-(SP) ATR_DESC		4027
		1C	7E 7E	D4 7C	0017B 00180		CLRL CLRW	-(SP) FIB_DESC		4033
		78	AE 7E	9F 7C	00184 00187		PUSHAB CLRW	-(SP) -(SP)		
		20	AE 32	9F DD	00189 0018C		PUSHAB PUSHL	IOSB #50		
	68		67 7E	DD D4	0018E 00190		PUSHL CLRL	OUTPUT_CHAN -(SP)		
56	03		0C 50	FB D0	00192 00195		CALLS MOVL	#12, SYSSQIOW R0, STATUS		4034
56			56 6E	E9 3C	00198 0019B		BLBC MOVZWL	STATUS, 13\$ IOSB, STATUS		
	69		67 01	DD FB	0019E 001A0	13\$:	PUSHL CALLS	OUTPUT_CHAN #1, SYSSDASSGN		4036
00000910	8F		67 56	D4 D1	001A3 001A5		CLRL Cmpl	OUTPUT_CHAN STATUS, #2320		4037
	10		33 56	13 E8	001AC 001AE		BEQL BLBS	16\$ STATUS, 15\$		4038
		08	56 A7	DD DD	001B1 001B3		PUSHL PUSHL	STATUS OUTPUT_FAB		4041
00000000G	00		5B 03	DD FB	001B6 001B8		PUSHL CALLS	R11 #3, FILE_ERROR		4044
			24 1C	11 13	001BF 001C7	14\$: 15\$:	BRB BISL3	17\$ BAKDATE+4, BAKDATE, R0		4045
50	10	AE	14	AE	C9	001C1	BEQL	17\$		4054
	FF40	C7	14	AE	D1	001C9	Cmpl	BAKDATE+4, COM_O_BSRDATE+4		4055
	FF40	C7	14	AE	1F	001CF	BLSSU	16\$		4056
	FF3C	C7	10	AE	D1	001D1	Cmpl	BAKDATE+4, COM_O_BSRDATE+4		
				0C	12	001D7	BNEQ	17\$		
				AF	D1	001D9	Cmpl	BAKDATE, COM_O_BSRDATE		
				04	1A	001DF	BGTRU	17\$		

RESTORE
V04-000

Restore a save set

SELECT_INCREMENTAL - apply incremental selectio

L 10

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742

[BACKUP.SRC]RESTORE.B32;1

Page 85

(16)

50

01 D0 001E1 16\$: MOVL #1, R0
04 001E4 RET
50 D4 001E5 17\$: CLRL R0
04 001E7 RET

; 4053

:

:

:

; 4061

; Routine Size: 488 bytes, Routine Base: CODE + 12A5

RESTORE
V04-000

Restore a save set
INCREMENTAL_DIRECTORY - perform incremental dir

M 10
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 86
(17)

```

: 2524 4062 1 %SBTTL 'INCREMENTAL_DIRECTORY - perform incremental directory scan'
: 2525 4063 1 ROUTINE INCREMENTAL_DIRECTORY(LENGTH,ADDRESS): NOVALUE=
: 2526 4064 1
: 2527 4065 1 ++
: 2528 4066 1
: 2529 4067 1 FUNCTIONAL DESCRIPTION:
: 2530 4068 1 This routine executes the special /INCREMENTAL directory scan.
: 2531 4069 1
: 2532 4070 1 INPUT PARAMETERS:
: 2533 4071 1 LENGTH - Length and address of directory segment.
: 2534 4072 1 ADDRESS -
: 2535 4073 1
: 2536 4074 1 IMPLICIT INPUTS:
: 2537 4075 1 NONE
: 2538 4076 1
: 2539 4077 1 OUTPUT PARAMETERS:
: 2540 4078 1 NONE
: 2541 4079 1
: 2542 4080 1 IMPLICIT OUTPUTS:
: 2543 4081 1 NONE
: 2544 4082 1
: 2545 4083 1 ROUTINE VALUE:
: 2546 4084 1 NONE
: 2547 4085 1
: 2548 4086 1 SIDE EFFECTS:
: 2549 4087 1 NONE
: 2550 4088 1
: 2551 4089 1 --
: 2552 4090 1
: 2553 4091 2 BEGIN
: 2554 4092 2 LOCAL
: 2555 4093 2 REC: REF BBLOCK, ! Pointer to directory record
: 2556 4094 2 NEXT_BLOCK; ! Pointer to next directory block
: 2557 4095 2
: 2558 4096 2
: 2559 4097 2 ! Initialize for the first block.
: 2560 4098 2
: 2561 4099 2 REC = .ADDRESS;
: 2562 4100 2 NEXT_BLOCK = .REC + 512;
: 2563 4101 2
: 2564 4102 2
: 2565 4103 2 ! Loop over all blocks.
: 2566 4104 2
: 2567 4105 2 WHILE .REC LSSA .ADDRESS + .LENGTH DO
: 2568 4106 3 BEGIN
: 2569 4107 3 IF .REC[DIR$W_SIZE] EQL 65535
: 2570 4108 3 THEN
: 2571 4109 4 BEGIN
: 2572 4110 4
: 2573 4111 4 ! End of this block. Advance to next and resume.
: 2574 4112 4
: 2575 4113 4 REC = .NEXT_BLOCK;
: 2576 4114 4 NEXT_BLOCK = .REC + 512;
: 2577 4115 4 END
: 2578 4116 3 ELSE
: 2579 4117 4 BEGIN
: 2580 4118 4 LOCAL
```


RESTORE
V04-000

Restore a save set
INCREMENTAL_DIRECTORY - perform incremental dir

N 10

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742

[BACKUP.SRC]RESTORE.B32;1

Page 87
(17)

```

: 2581      4119  4      NEXT_RECORD,      ! Pointer to next record
: 2582      4120  4      VER: REF BBLOCK;    ! Pointer to version entry
: 2583      4121  4
: 2584      4122  4
: 2585      4123  4      ! Point to where next record should start. Make some validity tests
: 2586      4124  4      ! on the entry we are looking at.
: 2587      4125  4
: 2588      4126  4      NEXT_RECORD = .REC[DIR$W_SIZE] + .REC + 2;
: 2589      4127  4      IF
: 2590      4128  5          BEGIN
: 2591      4129  5              IF
: 2592      4130  5                  .NEXT_RECORD GEQA .NEXT_BLOCK OR      ! Entry within block?
: 2593      4131  5                  .REC[DIR$W_SIZE] OR                  ! Length even?
: 2594      4132  5                  .REC[DIR$W_SIZE] LSSU DIR$C_LENGTH + DIR$C_VERSION ! Long enough?
: 2595      4133  5              THEN
: 2596      4134  5                  TRUE
: 2597      4135  5              ELSE
: 2598      4136  6                  BEGIN
: 2599      4137  6                      VER = (.REC + DIR$C_LENGTH + .REC[DIR$B_NAMECOUNT] + 1) AND NOT 1;
: 2600      4138  6                      .REC[DIR$V_TYPE] NEQ DIR$C_FID OR      ! Proper type code?
: 2601      4139  6                      .VER GEQA .NEXT_BLOCK - DIR$C_VERSION ! Version entry within block?
: 2602      4140  6                      END
: 2603      4141  5                  END
: 2604      4142  4              THEN
: 2605      4143  5                  BEGIN
: 2606      4144  5                      !
: 2607      4145  5                      ! Directory format error. Report it and quit.
: 2608      4146  5                      !
: 2609      4147  5                      FILE_ERROR(BACKUP$_INCBADDIR, .OUTPUT_FAB);
: 2610      4148  5                      RETJRN;
: 2611      4149  4                      END;
: 2612      4150  4
: 2613      4151  4      ! Loop over all version entries.
: 2614      4152  4      !
: 2615      4153  4      WHILE .VER LSSA .NEXT_RECORD DO
: 2616      4154  4          BEGIN
: 2617      4155  5              WHILE
: 2618      4156  5                  BEGIN
: 2619      4157  6                      IF .INCR_RSS EQL 0
: 2620      4158  6                      THEN
: 2621      4159  6                          FALSE
: 2622      4160  6                      ELSE
: 2623      4161  6                          BEGIN
: 2624      4162  7                              CH$LESS(
: 2625      4163  7                                  .INCR_NTS, INCR_RSA,
: 2626      4164  7                                  .REC[DIR$B_NAMECOUNT], REC[DIR$T_NAME]) OR
: 2627      4165  7                                  (CH$EQL(
: 2628      4166  8                                      .INCR_NTS, INCR_RSA,
: 2629      4167  8                                      .REC[DIR$B_NAMECOUNT], REC[DIR$T_NAME]) AND
: 2630      4168  8                                      .INCR_VER GTRU .VER[DIR$W_VERSION])
: 2631      4169  8                              END
: 2632      4170  7                          END
: 2633      4171  6                      END
: 2634      4172  5                  DO
: 2635      4173  6                      BEGIN
: 2636      4174  6                          INCREMENTAL_DELETE();
: 2637      4175  6                          INCREMENTAL_SCAN();
```

```

: 2638      4176 5      END;
: 2639      4177 5
: 2640      4178 5
: 2641      4179 5      IF .INCR_RSS NEQ 0
: 2642      4180 5      AND CH$EQL(
: 2643      4181 5          .INCR_NTS, INCR_RSA,
: 2644      4182 5          .REC[DIR$B_NAMECOUNT], REC[DIR$T_NAME]) AND
: 2645      4183 5          .INCR_VER EQL .VER[DIR$W_VERSION]
: 2646      4184 5      THEN
: 2647      4185 5          INCREMENTAL_SCAN()
: 2648      4186 5      ELSE
: 2649      4187 6          BEGIN
: 2650      4188 6              LOCAL
: 2651      4189 6                  FIB_DESC: VECTOR[2],
: 2652      4190 6                  FIB: BBLOCK[FIB$C_LENGTH],
: 2653      4191 6                  FNA_DESC: VECTOR[2],
: 2654      4192 6                  FNA: VECTOR[DIR$S_NAME+6,BYTE],
: 2655      4193 6                  RSA_DESC: VECTOR[2],
: 2656      4194 6                  RSA: VECTOR[NAM$C_MAXRSS,BYTE],
: 2657      4195 6                  STATUS,
: 2658      4196 6                  IOSB: VECTOR[4,WORD];
: 2659      4197 6
: 2660      4198 6          : Construct the name, type, and version string from the
: 2661      4199 6          : information in the directory entry.
: 2662      4200 6          FNA_DESC[0] = DIR$S_NAME + 6;
: 2663      4201 6          FNA_DESC[1] = FNA;
: 2664      4202 6          $FAD(
: 2665      4203 6              $DESCRIPTOR('!AC;!UW'),
: 2666      4204 6              FNA_DESC,
: 2667      4205 6              FNA_DESC,
: 2668      4206 6              REC[DIR$B_NAMECOUNT],
: 2669      4207 6              .VER[DIR$W_VERSION]);
: 2670      4208 6
: 2671      4209 6          : Construct a "resultant string" by concatenating the device
: 2672      4210 6          : and directory specification, which is in the name block
: 2673      4211 6          : expanded string, with the file name string.
: 2674      4212 6          RSA_DESC[0] = NAM$C_MAXRSS;
: 2675      4213 6          RSA_DESC[1] = RSA;
: 2676      4214 6          $FAD(
: 2677      4215 6              $DESCRIPTOR('!AD!AS'),
: 2678      4216 6              RSA_DESC,
: 2679      4217 6              RSA_DESC,
: 2680      4218 6              .OUTPUT_NAM[NAM$B_ESL], .OUTPUT_NAM[NAM$L_ESA],
: 2681      4219 6              FNA_DESC);
: 2682      4220 6
: 2683      4221 6          : Set up for and enter the directory entry.
: 2684      4222 6          FIB_DESC[0] = FIB$C_LENGTH;
: 2685      4223 6          FIB_DESC[1] = FIB;
: 2686      4224 6          CH$FILL(0, FIB$C_LENGTH, FIB);
: 2687      4225 6          FIB[FIB$W_FID_NUM] = -1;
: 2688      4226 6          FIB[FIB$W_FID_SEQ] = -1;
: 2689      4227 6
: 2690      4228 6
: 2691      4229 6
: 2692      4230 6
: 2693      4231 6
: 2694      4232 6
```

RESTORE
V04-000

Restore a save set
INCREMENTAL_DIRECTORY - perform incremental dir

C 11
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKLP.SRC]RESTORE.B32;1

Page 89
(17)

```

: 2695      4233 6      FIB[FIB$B_FID_NMX] = -1;
: 2696      4234 6      FIB[FIB$W_DID_NUM] = .INCR_FIB[FIB$W_DID_NUM];
: 2697      4235 6      FIB[FIB$W_DID_SEQ] = .INCR_FIB[FIB$W_DID_SEQ];
: 2698      4236 6      FIB[FIB$W_DID_RVN] = .INCR_FIB[FIB$W_DID_RVN];
: 2699      P 4237 6      STATUS = $QIOQ(
: 2700      P 4238 6      FUNC=IOS_CREATE,
: 2701      P 4239 6      CHAN=.OUTPUT_CHAN,
: 2702      P 4240 6      IOSB=IOSB,
: 2703      P 4241 6      P1=FIB_DESC,
: 2704      4242 6      P2=FNA_DESC);
: 2705      4243 6      IF .STATUS THEN STATUS = .IOSB[0];
: 2706      4244 6      IF NOT .STATUS
: 2707      4245 6      THEN
: 2708      4246 6      SIGNAL(BACKUPS_INCENTERR, 1, RSA_DESC, .STATUS);
: 2709      4247 5      END;
: 2710      4248 5
: 2711      4249 5      ! Advance to next version entry.
: 2712      4250 5      !
: 2713      4251 5      VER = .VER + DIR$C_VERSION;
: 2714      4252 5      END;
: 2715      4253 4
: 2716      4254 4
: 2717      4255 4
: 2718      4256 4      ! Advance to next directory record.
: 2719      4257 4      !
: 2720      4258 4      REC = .NEXT_RECORD;
: 2721      4259 3      END;
: 2722      4260 2      END;
: 2723      4261 1      END;
```

```

57 55 21 3B 43 41 21 0148D P.AAM: .ASCII \!AC;!UW\
      00000007 01494 P.AAL: .LONG 7
      00000000' 01498 .ADDRESS P.AAM
53 41 21 44 41 21 0149C P.AAO: .ASCII \!AD!AS\
      014A2 .BLKB 2
      00000006 014A4 P.AAN: .LONG 6
      00000000' 014A8 .ADDRESS P.AAO
```

```

OFFC 0000 INCREMENTAL DIRECTORY:
      5B 00000000' EF 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 4063
      5E FE48 CE 9E 00009 MOVAB INCR_NIS, R11
      56 08 AC D0 0000E MOVAB -440(SP), SP
      57 0200 C6 9E 00012 MOVL ADDRESS, REC : 4099
SA 08 AC C1 00017 MOVAB 512(R6), NEXT_BLOCK : 4100
SA 04 56 D1 0001D 1$: ADDL3 LENGTH, ADDRESS, R10 : 4105
      01 1F 00020 CMPL REC, R10
      04 00022 BLSSU 2$
      50 66 3C 00023 2$: RET
FFFF 8F 50 B1 00026 MOVZWL (REC), R0 : 4107
      0A 12 0002B CMPW R0, #65535
      56 57 D0 0002D BNEQ 3$
      57 0200 C6 9E 00030 MOVL NEXT_BLOCK, REC : 4113
      MOVAB 512(R6), NEXT_BLOCK : 4114
```

				59	02	A6	40	E6	11	00035	BRB	1\$	4107
				57				9E	00037	3\$:	MOVAB	2(REC)[R0], NEXT_RECORD	4126
								D1	0003C		CMPL	NEXT_RECORD, NEXT_BLOCK	4130
								24	1E	0003F	BGEQU	4\$	
				21				50	E8	00041	BLBS	R0, 4\$	4131
				0E				50	B1	00044	CMPL	R0, #14	4132
								1C	1F	00047	BLSSU	4\$	
				50	05	A6	9A	9A	00049		MOVZBL	5(REC), R0	4137
				50	07	A0	46	9E	0004D		MOVAB	7(R0)[REC], R0	
58				50		01	CB	00052			BICL3	#1, R0, VER	
				07	04	A6	93	00056			BITB	4(REC), #7	4138
						09	12	0005A			BNEQ	4\$	
				50	F8	A7	9E	0005C			MOVAB	-8(R7), R0	4139
				50		58	D1	00060			CMPL	VER, R0	
						14	1F	00063			BLSSU	5\$	
						EF	DD	00065	4\$:		PUSHL	OUTPUT_FAB	4147
						8F	DD	0006B			PUSHL	#BACKUP\$_INCBADDR	
						02	FB	00071			CALLS	#2, FILE_ERROR	
								04	00078		RET		4143
				59		58	D1	00079	5\$:		CMPL	VER, NEXT_RECORD	4154
						03	1F	0007C			BLSSU	6\$	
						01	15	31	0007E		BRW	12\$	
					FC	AB	D5	00081	6\$:		TSTL	INCR_RSS	4158
						30	13	00084			BEQL	8\$	
				50	05	A6	9A	00086			MOVZBL	5(REC), R0	4165
						6B	2D	0008A			CMPC5	INCR_NTS, INCR_RSA, #0, R0, 6(REC)	
					06	A6		00090					
						16	1F	00092			BLSSU	7\$	
				50	05	A6	9A	00094			MOVZBL	5(REC), R0	4168
						6B	2D	00098			CMPC5	INCR_NTS, INCR_RSA, #0, R0, 6(REC)	
					06	A6		0009E					
						14	12	000A0			BNEQ	8\$	
						00	EC	000A2			CMPL	#0, #16, (VER), INCR_VER	4169
						0C	1E	000A8			BGEQU	8\$	
						00	FB	000AA	7\$:		CALLS	#0, INCREMENTAL_DELETE	4174
						00	FB	000AF			CALLS	#0, INCREMENTAL_SCAN	4175
						CB	11	000B4			BRB	6\$	4156
					FC	AB	D5	000B6	8\$:		TSTL	INCR_RSS	4179
						1E	13	000B9			BEQL	9\$	
				50	05	A6	9A	000BB			MOVZBL	5(REC), R0	4182
						6B	2D	000BF			CMPC5	INCR_NTS, INCR_RSA, #0, R0, 6(REC)	
					06	A6		000C5					
						10	12	000C7			BNEQ	9\$	
						00	EC	000C9			CMPL	#0, #16, (VER), INCR_VER	4183
						08	12	000CF			BNEQ	9\$	
						00	FB	000D1			CALLS	#0, INCREMENTAL_SCAN	4185
						00	B7	31	000D6		BRW	11\$	
						8F	9A	000D9	9\$:		MOVZBL	#86, FNA_DESC	4202
						CD	9E	000DE			MOVAB	FNA, FNA_DESC+4	4203
						68	32	000E4			CVTL	(VER), -(TSP)	4209
					05	A6	9F	000E7			PUSHAB	5(REC)	
					B0	AD	9F	000EA			PUSHAB	FNA_DESC	
					B0	AD	9F	000ED			PUSHAB	FNA_DESC	
					FEF4	CF	9F	000F0			PUSHAB	P.AXL	
						05	FB	000F4			CALLS	#5, SYSSFA0	
						FF	8F	9A	000FB		MOVZBL	#255, RSA_DESC	4216
						08	AE	9E	00101		MOVAB	RSA, RSA_DESC+4	4217

RESTORE
V04-000

Restore a save set
INCREMENTAL_DIRECTORY - perform incremental dir

E 11
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 91
(17)

			B0	AD	9F	00107	PUSHAB	FNA_DESC		4223
		50	00000000	EF	DD	0010A	MOVL	OUTPUT_NAM, R0		
			0C	AO	DD	00111	PUSHL	12(R0)		
		7E		0B	AO	9A 00114	MOVZBL	11(R0), -(SP)		
			FF50	CD	9F	00118	PUSHAB	RSA_DESC		
			FF50	CD	9F	0011C	PUSHAB	RSA_DESC		
			FED4	CF	9F	00120	PUSHAB	P.AAN		
		00000000G	00	06	FB	00124	CALLS	#6, SYSSFAO		
		F8	AD	40	8F	9A 0012B	MOVZBL	#64, FIB_DESC		4228
		FC	AD	B8	AD	9E 00130	MOVAB	FIB, FIB_DESC+4		4229
0040	8f	00	6E	00	2C	00135	MOVCS	#0, (SP); #0, #64, FIB		4230
			B8	AD		0013C				
		BC	AD	01	CE	0013E	MNEGL	#1, FIB+4		4231
		C1	AD	01	8E	00142	MNEGB	#1, FIB+9		4233
		C2	AD	FF6E	CB	DD 00146	MOVL	INCR FIB+10, FIB+10		4234
		C6	AD	FF72	CB	DD 0014C	MOVW	INCR FIB+14, FIB+14		4236
				7E	7C	00152	CLRQ	-(SP)		4242
				7E	7C	00154	CLRQ	-(SP)		
			B0	AD	9F	00156	PUSHAB	FNA_DESC		
			F8	AD	9F	00159	PUSHAB	FIB_DESC		
				7E	7C	0015C	CLRQ	-(SP)		
			20	AE	9F	0015E	PUSHAB	IOSB		
				33	DD	00161	PUSHL	#51		
			00000000	EF	DD	00163	PUSHL	OUTPUT_CHAN		
				7E	D4	00169	CLRL	-(SP)		
		00000000G	00	0C	FB	0016B	CALLS	#12, SYSSQIOW		
			06	50	E9	00172	BLBC	STATUS, 10\$		4243
			50	6E	3C	00175	MOVZWL	IOSB, STATUS		
			15	50	E8	00178	BLBS	STATUS, 11\$		4244
				50	DD	0017B	PUSHL	STATUS		4246
			FF50	CD	9F	0017D	PUSHAB	RSA_DESC		
				01	DD	00181	PUSHL	#1		
			00000000G	8F	DD	00183	PUSHL	#BACKUPS INCENTERR		
		00000000G	00	04	FB	00189	CALLS	#4, LIB\$SIGNAL		
			58	08	CD	00190	ADDL2	#8, VER		4252
				FEE3	31	00193	BRW	5\$		4154
			56	59	DD	00196	MOVL	NEXT_RECORD, REC		4258
				FE81	31	00199	BRW	1\$		4105
				04	0019C		RET			4261

; Routine Size: 413 bytes, Routine Base: CODE + 14AC

```
: 2725 4262 1 %SBTTL 'CREATE_OR_OPEN_FILE - create or open output file'
: 2726 4263 1 ROUTINE CREATE_OR_OPEN_FILE: NOVALUE=
: 2727 4264 1
: 2728 4265 1 !++
: 2729 4266 1
: 2730 4267 1 FUNCTIONAL DESCRIPTION:
: 2731 4268 1 This routine creates or opens a file in response to
: 2732 4269 1 a file attribute record.
: 2733 4270 1
: 2734 4271 1 INPUT PARAMETERS:
: 2735 4272 1 NONE
: 2736 4273 1
: 2737 4274 1 IMPLICIT INPUTS:
: 2738 4275 1 NONE
: 2739 4276 1
: 2740 4277 1 OUTPUT PARAMETERS:
: 2741 4278 1 NONE
: 2742 4279 1
: 2743 4280 1 IMPLICIT OUTPUTS:
: 2744 4281 1 NONE
: 2745 4282 1
: 2746 4283 1 ROUTINE VALUE:
: 2747 4284 1 NONE
: 2748 4285 1
: 2749 4286 1 SIDE EFFECTS:
: 2750 4287 1 NONE
: 2751 4288 1
: 2752 4289 1 --
: 2753 4290 1
: 2754 4291 2 BEGIN
: 2755 4292 2
: 2756 4293 2 ! Compute the size of the saved file and set the expected highwater
: 2757 4294 2 mark to match.
: 2758 4295 2
: 2759 4296 2 OUTPUT_BLOCK = 1;
: 2760 4297 2 OUTPUT_MAXBLOCK = ROT(.OUTPUT_RECATTR[FAT$L_EFBLK], 16);
: 2761 4298 2 IF .OUTPUT_RECATTR[FAT$W_FFBYTE] EQL 0
: 2762 4299 2 THEN OUTPUT_MAXBLOCK = .OUTPUT_MAXBLOCK - 1;
: 2763 4300 2 IF
: 2764 4301 2 .OUTPUT_MAXBLOCK GTRU .OUTPUT_ATTBUF[FAR_FILESIZE] OR
: 2765 4302 2 .OUTPUT_RECATTR[FAT$V_FILEORG] NEQ FAT$C_SEQUENTIAL
: 2766 4303 2 THEN
: 2767 4304 2 OUTPUT_MAXBLOCK = .OUTPUT_ATTBUF[FAR_FILESIZE];
: 2768 4305 2 OUTPUT_ATTBUF[FAR_HIGHWATER] = .OUTPUT_MAXBLOCK + 1;
: 2769 4306 2
: 2770 4307 2
: 2771 4308 2 IF .COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]
: 2772 4309 2 THEN
: 2773 4310 3 BEGIN
: 2774 4311 3 LOCAL
: 2775 4312 3 FIB: BBLOCK[FIB$C_LENGTH], ! FIB
: 2776 4313 3 FIB_DESC: VECTOR[2], ! Descriptor for FIB
: 2777 4314 3 STATUS, ! Status variable
: 2778 4315 3 IOSB: VECTOR[4,WORD]; ! I/O status block
: 2779 4316 3
: 2780 4317 3
: 2781 4318 3 ! Construct a "resultant string" from the device name that is present in
```

```
2782 4319 3 ! the expanded string of the current verification qualifier block and the
2783 4320 3 ! filename string from the save set.
2784 4321 3
2785 4322 3 OUTPUT_NAM[NAM$B_RSL] =
2786 4323 3 .BBLOCK[VERIFY_QUAL[QUAL_DEV_DESC], DSC$W_LENGTH] +
2787 4324 3 .OUTPUT_FILENAME[DSC$W_LENGTH];
2788 4325 3 CH$MOVE(
2789 4326 3 .OUTPUT_FILENAME[DSC$W_LENGTH],
2790 4327 3 .OUTPUT_FILENAME[DSC$A_POINTER],
2791 4328 3 CH$MOVE(
2792 4329 3 .BBLOCK[VERIFY_QUAL[QUAL_DEV_DESC], DSC$W_LENGTH],
2793 4330 3 .BBLOCK[VERIFY_QUAL[QUAL_DEV_DESC], DSC$A_POINTER],
2794 4331 3 .OUTPUT_NAM[NAM$R_RSA]));
2795 4332 3 INIT_NAMEBLOCK(.OUTPUT_NAM);
2796 4333 3
2797 4334 3
2798 4335 3 ! Initialize the FIB with the file ID from the save set,
2799 4336 3 ! and access the file.
2800 4337 3
2801 4338 3 CH$FILL (0, FIB$C_LENGTH, FIB);
2802 4339 3 FIB[FIB$L_ACCTL] = FIB$M_NOWRITE;
2803 4340 3 IF
2804 4341 3 .QUAL[QUAL_IGNORE_INT] OR
2805 4342 4 (.OUTPUT_FILENAME[DSC$W_LENGTH] EQL %CHARCOUNT('[000000]QUOTA.SYS;1') AND
2806 4343 4 CH$EQL(
2807 4344 4 %CHARCOUNT('[000000]QUOTA.SYS;1'),
2808 4345 4 UPLIT_BYTE('[000000]QUOTA.SYS;1'),
2809 4346 4 %CHARCOUNT('[000000]QUOTA.SYS;1'),
2810 4347 4 .OUTPUT_FILENAME[DSC$A_POINTER]))
2811 4348 3 THEN
2812 4349 3 FIB[FIB$L_ACCTL] = FIB$M_NOLOCK;
2813 4350 3 FIB[FIB$V_NORECORD] = TRUE;
2814 4351 3 FIB[FIB$W_FID_NUM] = .OUTPUT_ATTBUF[FAR_FID_NUM];
2815 4352 3 FIB[FIB$W_FID_SEQ] = .OUTPUT_ATTBUF[FAR_FID_SEQ];
2816 4353 3 FIB[FIB$W_FID_RVN] = .OUTPUT_ATTBUF[FAR_FID_RVNW];
2817 4354 3 FIB_DESC[0] = FIB$C_LENGTH;
2818 4355 3 FIB_DESC[1] = FIB;
2819 4356 3 STATUS = $QIOW(
2820 4357 3 FUNC=IOS_ACCESS OR IOSM_ACCESS,
2821 4358 3 CHAN=.OUTPUT_CHAN,
2822 4359 3 IOSB=IOSB,
2823 4360 3 P1=FIB_DESC);
2824 4361 3 IF .STATUS THEN STATUS = .IOSB[0];
2825 4362 3 IF NOT .STATUS
2826 4363 3 THEN
2827 4364 4 BEGIN
2828 4365 4 FILE_ERROR(
2829 4366 4 BACKUP$OPENIN + ST$K_ERROR,
2830 4367 4 .OUTPUT_FAB,
2831 4368 4 .STATUS);
2832 4369 4 RETURN;
2833 4370 3 END;
2834 4371 3 END
2835 4372 2 ELSE IF .QUAL[QUAL_IMAG]
2836 4373 2 THEN
2837 4374 3 BEGIN
2838 4375 3 LOCAL
```

```

2839      4376      3      FIB:      BBLOCK[FIB$C_LENGTH],      ! FIB
2840      4377      3      FIB_DESC:      VECTOR[2],      ! Descriptor for FIB
2841      4378      3      STATUS,      ! Status variable
2842      4379      3      IOSB:      VECTOR[4,WORD];      ! I/O status block
2843      4380      3
2844      4381      3
2845      4382      3      ! Construct a 'resultant string' from the device name that is present in
2846      4383      3      ! the expanded string of the output qualifier block and the filename
2847      4384      3      ! string from the save set.
2848      4385      3
2849      4386      3      OUTPUT_NAM[NAM$B_RSL] = .OUTPUT_NAM[NAM$B_ESL] + .OUTPUT_FILENAME[DSC$W_LENGTH];
2850      4387      3      CH$MOVE(
2851      4388      3          .OUTPUT_FILENAME[DSC$W_LENGTH],
2852      4389      3          .OUTPUT_FILENAME[DSC$A_POINTER],
2853      4390      3          CH$MOVE(
2854      4391      3              .OUTPUT_NAM[NAM$B_ESL],
2855      4392      3              .OUTPUT_NAM[NAM$B_ESA],
2856      4393      3              .OUTPUT_NAM[NAM$B_RSA]));
2857      4394      3      INIT_NAMEBLOCK(.OUTPUT_NAM);
2858      4395      3
2859      4396      3
2860      4397      3      ! Set the file ID into the NAM block.
2861      4398      3
2862      4399      3      OUTPUT_NAM[NAM$W_FID_NUM] = .OUTPUT_ATTBUF[FAR_FID_NUM];
2863      4400      3      OUTPUT_NAM[NAM$W_FID_SEQ] = .OUTPUT_ATTBUF[FAR_FID_SEQ];
2864      4401      3      OUTPUT_NAM[NAM$W_FID_RVN] = .OUTPUT_ATTBUF[FAR_FID_RVNW];
2865      4402      3
2866      4403      3
2867      4404      3      ! Initialize the FIB. Make sure the file ID is non-zero.
2868      4405      3
2869      4406      3      CH$FILL(0, FIB$C_LENGTH, FIB);
2870      4407      3      FIB[FIB$W_FID_NUM] = .OUTPUT_ATTBUF[FAR_FID_NUM];
2871      4408      3      FIB[FIB$W_FID_SEQ] = .OUTPUT_ATTBUF[FAR_FID_SEQ];
2872      4409      3      FIB[FIB$W_FID_RVN] = .OUTPUT_ATTBUF[FAR_FID_RVNW];
2873      4410      3      FIB_DESC[0] = FIB$C_LENGTH;
2874      4411      3      FIB_DESC[1] = FIB;
2875      4412      3
2876      4413      3      IF .FIB[FIB$W_FID_NUM] EQL 0
2877      4414      3      AND .FIB[FIB$W_FID_RVN] EQL 0
2878      4415      3      THEN SIGNAL (BACKUP$_INVFID, 0);
2879      4416      3
2880      4417      3
2881      4418      3      IF .QUAL[QUAL_COMP]
2882      4419      3      THEN
2883      4420      4          BEGIN
2884      4421      4
2885      4422      4          ! Compare operation: Access the file by its file ID.
2886      4423      4
2887      4424      4          FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
2888      4425      4          IF
2889      4426      4              .QUAL[QUAL_IGNORE_INT] OR
2890      4427      5              (.OUTPUT_FILENAME[DSC$W_LENGTH] EQL %CHARCOUNT('[000000]QUOTA.SYS;1') AND
2891      4428      5              CH$EQL(
2892      4429      5                  %CHARCOUNT('[000000]QUOTA.SYS;1'),
2893      4430      5                  UPLIT_BYTE('[000000]QUOTA.SYS;1'),
2894      4431      5                  %CHARCOUNT('[000000]QUOTA.SYS;1'),
2895      4432      5                  .OUTPUT_FILENAME[DSC$A_POINTER]))

```



```
2896 4433 4 THEN
2897 4434 4     FIB[FIB$L ACCTL] = FIB$M_NOLOCK;
2898 4435 4     FIB[FIB$V NORECORD] = TRUE;
2899 P 4436 4     STATUS = [$QIOW(
2900 P 4437 4         FUNC=IOS_ACCESS OR IOSM_ACCESS,
2901 P 4438 4         CHAN=.OUTPUT_CHAN,
2902 P 4439 4         IOSB=IOSB,
2903 4440 4         P1=FIB_DE$C);
2904 4441 4     IF .STATUS THEN STATUS = .IOSB[0];
2905 4442 4     IF NOT .STATUS
2906 4443 4     THEN
2907 4444 5         BEGIN
2908 4445 5             FILE_ERROR(
2909 4446 5                 BACKUP$_OPENIN + ST$K_ERROR,
2910 4447 5                 .OUTPUT_FAB,
2911 4448 5                 .STATUS);
2912 4449 5             RETURN;
2913 4450 4         END;
2914 4451 4     ELSE
2915 4452 3     BEGIN
2916 4453 4         LOCAL
2917 4454 4             ATR_LIST:  VECTOR[MAX_ATTRS*2 + 1]; ! ACP attributes list
2918 4455 4
2919 4456 4         GET_ACP_ATTRIBUTES(ATR_LIST);
2920 4457 4         IF
2921 4458 4             (.OUTPUT_ATTBUF[FAR FID_NMX] EQL 0 AND
2922 4459 4             (.OUTPUT_ATTBUF[FAR FID_NUM] LEQU FID$C_BADBLK OR
2923 4460 4             (.OUTPUT_ATTBUF[FAR-STRUCLEVB] EQL 2 AND
2924 4461 5             .OUTPUT_ATTBUF[FAR_FID_NUM] EQL FID$C_BADLOG))
2925 4462 6         THEN
2926 4463 5             BEGIN
2927 4464 4                 LOCAL
2928 4465 5                     OLD_RECATTR:  BBLOCK[FAT$C_LENGTH],
2929 4466 5                     FAT_ATR_LIST:  BBLOCK[12];
2930 4467 5
2931 4468 5                     ! File attribute record for an initial file. Read the record
2932 4469 5                     ! attributes. This is required because HIBLK and EFBLK are
2933 4470 5                     ! incorrect in the save set record attributes.
2934 4471 5                     FAT_ATR_LIST[0,0,16,0] = ATR$S_RECATTR;
2935 4472 5                     FAT_ATR_LIST[2,0,16,0] = ATR$C_RECATTR;
2936 4473 5                     FAT_ATR_LIST[4,0,32,0] = OLD_RECATTR;
2937 4474 5                     FAT_ATR_LIST[8,0,32,0] = 0;
2938 4475 5                     STATUS = $QIOW(
2939 4476 5                         FUNC=IOS_ACCESS,
2940 4477 5                         CHAN=.OUTPUT_CHAN,
2941 4478 5                         IOSB=IOSB,
2942 P 4479 5                         P1=FIB_DE$C,
2943 P 4480 5                         PS=FAT_ATR_LIST);
2944 P 4481 5                     IF .STATUS THEN STATUS = .IOSB[0];
2945 P 4482 5                     IF NOT .STATUS
2946 P 4483 5                     THEN
2947 4484 5                         BEGIN
2948 4485 5                             FILE_ERROR(
2949 4486 5                                 BACKUP$_READATTR,
2950 4487 5                                 .STATUS);
2951 4488 5                         END;
2952 4489 5                     RETURN;
```

```
2953      4490      5      .OUTPUT_FAB,  
2954      4491      5      .STATUS);  
2955      4492      5  
2956      4493      5  
2957      4494      5      ! Copy HIBLK and EFBLK to the new record attributes.  
2958      4495      5      !  
2959      4496      5      OUTPUT_RECATTR[FAT$HIBLK] = .OLD_RECATTR[FAT$HIBLK];  
2960      4497      5      OUTPUT_RECATTR[FAT$EFBLK] = .OLD_RECATTR[FAT$EFBLK];  
2961      4498      5  
2962      4499      5  
2963      4500      5      ! Write the new record attributes.  
2964      4501      5      !  
2965      P 4502      5      STATUS = $$QIOW(  
2966      P 4503      5      FUNC=IOS_MODIFY,  
2967      P 4504      5      CHAN=.OUTPUT_CHAN,  
2968      P 4505      5      IOSB=IOSB,  
2969      P 4506      5      P1=FIB_DESC,  
2970      P 4507      5      P5=ATR_LIST,  
2971      4508      5      P6=OUTPUT_ATTBUF);  
2972      4509      5      IF .STATUS THEN STATUS = .IOSB[0];  
2973      4510      5      IF NOT .STATUS  
2974      4511      5      THEN  
2975      4512      5          FILE_ERROR(  
2976      4513      5              BACKUP$WRITEATTR,  
2977      4514      5              .OUTPUT_FAB,  
2978      4515      5              .STATUS);  
2979      4516      5      RETURN;  
2980      4517      5      END  
2981      4518      4      ELSE  
2982      4519      5      BEGIN  
2983      4520      5      ! Create the file.  
2984      4521      5      !  
2985      4522      5      LOCAL  
2986      4523      5          NAM_DESC:      VECTOR[2];      ! Descriptor for P2 parameter  
2987      4524      5  
2988      4525      5  
2989      4526      5      ! Get the string that will go into the file header by  
2990      4527      5      ! stripping the directory specification.  
2991      4528      5      !  
2992      4529      5      !  
2993      4530      5      NAM_DESC[0] = .OUTPUT_NAM[NAM$RSA] + .OUTPUT_NAM[NAM$B_RSL] - .OUTPUT_NAM[NAM$NAME];  
2994      4531      5      NAM_DESC[1] = .OUTPUT_NAM[NAM$NAME];  
2995      4532      5  
2996      4533      5  
2997      4534      5      ! Initialize FIB fields needed for create.  
2998      4535      5      !  
2999      4536      5      FIB[FIB$W_DID_NUM] = .OUTPUT_ATTBUF[FAR_DID_NUM];  
3000      4537      5      FIB[FIB$W_DID_SEQ] = .OUTPUT_ATTBUF[FAR_DID_SEQ];  
3001      4538      5      FIB[FIB$W_DID_RVN] = .OUTPUT_ATTBUF[FAR_DID_RVNW];  
3002      4539      5      FIB[FIB$V_EXTEND] = TRUE;  
3003      4540      5      IF .OUTPUT_FILECHAR[FCH$V_CONTIG] THEN FIB[FIB$V_ALCON] = TRUE;  
3004      4541      5      FIB[FIB$L_EXSZ] = .OUTPUT_ATTBUF[FAR_FILESIZE];  
3005      4542      5  
3006      4543      5  
3007      4544      5      ! Pass a create QIO to the standalone ACP.  
3008      4545      5      !  
3009      P 4546      5      STATUS = $$QIOW(
```

RESTORE
V04-000

Restore a save set
CREATE_OR_OPEN_FILE - create or open output fil

K 11
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 97
(18)

```

3010 P 4547 5      FUNC=IOS$ CREATE OR IOSM_CREATE OR IOSM_ACCESS,
3011 P 4548 5      CHAN=.OUTPUT_CHAN,
3012 P 4549 5      IOSB=IOSB,
3013 P 4550 5      P1=FIB_DESC,
3014 P 4551 5      P2=NAM_DESC,
3015 P 4552 5      P5=ATR_LIST,
3016   4553 5      P6=OUTPUT_ATTBUF);
3017   4554 5      IF .STATUS THEN STATUS = .IOSB[0];
3018   4555 5
3019   4556 5
3020   4557 5      IF .STATUS EQL SS$_DEVICEFULL AND .FIB[FIB$_ALCON]
3021   4558 5      THEN
3022   4559 6      BEGIN
3023   4560 6
3024   4561 6      ! If a contiguous create failed, try a non-contiguous one.
3025   4562 6      !
3026   4563 6      FIB[FIB$_ALCON] = FALSE;
3027 P 4564 6      STATUS = SS$_IOW(
3028 P 4565 6      FUNC=IOS$ CREATE OR IOSM_CREATE OR IOSM_ACCESS,
3029 P 4566 6      CHAN=.OUTPUT_CHAN,
3030 P 4567 6      IOSB=IOSB,
3031 P 4568 6      P1=FIB_DESC,
3032 P 4569 6      P2=NAM_DESC,
3033 P 4570 6      P5=ATR_LIST,
3034   4571 6      P6=OUTPUT_ATTBUF);
3035   4572 6      IF .STATUS THEN STATUS = .IOSB[0];
3036   4573 6      IF .STATUS THEN FILE_ERROR(BACKUP$_NONCONTIG, .OUTPUT_FAB);
3037   4574 5      END;
3038   4575 5
3039   4576 5
3040   4577 5      IF NOT .STATUS
3041   4578 5      THEN
3042   4579 6      BEGIN
3043   4580 6      FILE_ERROR(
3044   4581 6      BACKUP$_OPENOUT + STS$_K_ERROR,
3045   4582 6      .OUTPUT_FAB,
3046   4583 6      .STATUS);
3047   4584 6      RETURN;
3048   4585 5      END;
3049   4586 5
3050   4587 5
3051   4588 5      ! If this is the boot file, rewrite the boot block with the
3052   4589 5      ! appropriate LBN.
3053   4590 5      !
3054   4591 5      IF .OUTPUT_ATTBUF[FAR_BOOTVBN] NEQ 0
3055   4592 5      THEN
3056   4593 5      STA_WRITEBOOT();
3057   4594 4      END;
3058   4595 3      END;
3059   4596 3
3060   4597 2      ELSE IF .QUAL[QUAL_COMP]
3061   4598 2      THEN
3062   4599 3      BEGIN
3063   4600 4      IF NOT $OPEN(FAB=.OUTPUT_FAB)
3064   4601 3      THEN
3065   4602 4      BEGIN
3066   4603 4      FILE_ERROR(
```

RESTORE
V04-000

Restore a save set

CREATE_OR_OPEN_FILE - create or open output fil

L 11
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 98
(18)

```

: 3067      4604 4      BACKUP$_OPENIN + ST$K_ERROR,
: 3068      4605 4      .OUTPUT_FAB,
: 3069      4606 4      .OUTPUT_FAB[FAB$L_STS], .OUTPUT_FAB[FAB$L_STV]);
: 3070      4607 4      RETURN;
: 3071      4608 4      END;
: 3072      4609 3      OUTPUT_CHAN = .OUTPUT_FAB[FAB$L_STV];
: 3073      4610 3      END
: 3074      4611 2      ELSE
: 3075      4612 3      BEGIN
: 3076      4613 3      LOCAL
: 3077      4614 3      SAVED_ALQ,          ! Saved allocation
: 3078      4615 3      SAVED_CTG,          ! Saved contiguous attribute
: 3079      4616 3      XABPRO:      $XABPRO_DECL,
: 3080      4617 3      STATUS;
: 3081      4618 3
: 3082      4619 3
: 3083      4620 3      ! If directory not found during $PARSE, try to create it.
: 3084      4621 3
: 3085      4622 3      IF .OUTPUT_FAB[FAB$L_STS] EQL RM$$_DNF
: 3086      4623 3      THEN
: 3087      4624 3          IF NOT IMPLICIT_DIRECTORY()
: 3088      4625 3          THEN
: 3089      4626 3              RETURN;
: 3090      4627 3
: 3091      4628 3
: 3092      4629 3      ! Establish ownership of the new file.
: 3093      4630 3
: 3094      4631 3      IF NOT .QUAL[QUAL_O_OWN_ORIG] AND NOT .QUAL[QUAL_INCR]
: 3095      4632 3      THEN
: 3096      4633 3          OUTPUT_ATTBUF[FAR_UIC] = .JPI_UIC;
: 3097      4634 3
: 3098      4635 3      IF .QUAL[QUAL_O_OWN_UIC]
: 3099      4636 3      THEN
: 3100      4637 3          OUTPUT_ATTBUF[FAR_UIC] = .QUAL[QUAL_O_OWN_VALU];
: 3101      4638 3
: 3102      4639 3      IF .QUAL[QUAL_O_OWN_PARE]
: 3103      4640 3      THEN
: 3104      4641 3          OUTPUT_ATTBUF[FAR_UIC] = GET_PARENT_UIC();
: 3105      4642 3
: 3106      4643 3
: 3107      4644 3      ! Initialize protection XAB for the file.
: 3108      4645 3
: 3109      4646 3      $XABPRO INIT(XAB=XABPRO);
: 3110      4647 3      XABPRO[XAB$L_UIC] = .OUTPUT_ATTBUF[FAR_UIC];
: 3111      4648 3      XABPRO[XAB$W_PRO] = $X'FFFF';          ! Use default protection
: 3112      4649 3      OUTPUT_FAB[FAB$L_XAB] = XABPRO;
: 3113      4650 3
: 3114      4651 3
: 3115      4652 3      ! Attempt to create the file.
: 3116      4653 3
: 3117      4654 3      SAVED_ALQ = .OUTPUT_FAB[FAB$L_ALQ];
: 3118      4655 3      SAVED_CTG = .OUTPUT_FAB[FAB$L_CTG];
: 3119      4656 3      STATUS = $CREATE(FAB=.OUTPUT_FAB);
: 3120      4657 3
: 3121      4658 3      ! If the create returns with an FNF error, we may have a dangling
: 3122      4659 3      ! directory entry in the directory path. Attempt to create it
: 3123      4660 3      ! and try again.
```

```

3124 4661 3 !
3125 4662 3 IF .STATUS EQL RMSS_FNF
3126 4663 3 THEN
3127 4664 4 BEGIN
3128 4665 4 IF NOT IMPLICIT_DIRECTORY () THEN RETURN;
3129 4666 4 STATUS = $CREATE (FAB = .OUTPUT_FAB);
3130 4667 3 END;
3131 4668 3
3132 4669 3
3133 4670 3 IF .STATUS EQL RMSS_FUL AND .OUTPUT_FAB[FAB$V_CTG]
3134 4671 3 THEN
3135 4672 4 BEGIN
3136 4673 4 ! If a contiguous create failed, try a non-contiguous one.
3137 4674 4 !
3138 4675 4 !
3139 4676 4 OUTPUT_FAB[FAB$V_CTG] = FALSE;
3140 4677 4 STATUS = $CREATE(FAB=.OUTPUT_FAB);
3141 4678 4 IF .STATUS THEN FILE_ERROR(BACKUP$NONCONTIG, .OUTPUT_FAB);
3142 4679 3 END;
3143 4680 3
3144 4681 3
3145 4682 3 OUTPUT_FAB[FAB$L_XAB] = 0;
3146 4683 3
3147 4684 3
3148 4685 3 IF NOT .STATUS
3149 4686 3 THEN
3150 4687 4 BEGIN
3151 4688 4 FILE_ERROR(
3152 4689 4 BACKUP$OPENOUT + STSSK_ERROR,
3153 4690 4 .OUTPUT_FAB,
3154 4691 4 .OUTPUT_FAB[FAB$L_STS], .OUTPUT_FAB[FAB$L_STV]);
3155 4692 4 RETURN;
3156 4693 3 END;
3157 4694 3
3158 4695 3
3159 4696 3 OUTPUT_CHAN = .OUTPUT_FAB[FAB$L_STV];
3160 4697 3
3161 4698 3
3162 4699 3 ! If the file is being created /OVERLAY and has insufficient space,
3163 4700 3 ! it must be extended.
3164 4701 3 !
3165 4702 3 IF .QUAL[QUAL_OVER] AND .OUTPUT_FAB[FAB$L_ALQ] LSSU .SAVED_ALQ
3166 4703 3 THEN
3167 4704 4 BEGIN
3168 4705 4 LOCAL
3169 4706 4 FIB: BBLOCK[FIB$C_LENGTH], ! FIB
3170 4707 4 FIB_DESC: VECTOR[2], ! Descriptor for FIB
3171 4708 4 STATUS, ! Status variable
3172 4709 4 IOSB: VECTOR[4,WORD]; ! I/O status block
3173 4710 4
3174 4711 4
3175 4712 4 CH$FILL(0, FIB$C_LENGTH, FIB);
3176 4713 4 FIB[FIB$V_NORECORD] = TRUE;
3177 4714 4 FIB[FIB$V_EXTEND] = TRUE;
3178 4715 4 FIB[FIB$L_EXSZ] = .SAVED_ALQ - .OUTPUT_FAB[FAB$L_ALQ];
3179 4716 4 FIB_DESC[0] = FIB$C_LENGTH;
3180 4717 4 FIB_DESC[1] = FIB;
```

```

: 3181 P 4718 4 STATUS = $QIOW(
: 3182 P 4719 4 FUNC=IOS MODIFY,
: 3183 P 4720 4 CHAN=.OUTPUT_CHAN,
: 3184 P 4721 4 IOSB=IOSB,
: 3185 4722 4 P1=FIB_DESC),
: 3186 4723 4 IF .STATUS THEN STATUS = .IOSB[0];
: 3187 4724 4 IF NOT .STATUS
: 3188 4725 4 THEN
: 3189 4726 5 BEGIN
: 3190 4727 5 FILE_ERROR(
: 3191 4728 5 BACKUP$ OPENOUT + ST$K_ERROR,
: 3192 4729 5 .OUTPUT_FAB,
: 3193 4730 5 .STATUS);
: 3194 4731 5 $DASSGN(CHAN=.OUTPUT_CHAN);
: 3195 4732 5 RETURN;
: 3196 4733 4 END;
: 3197 4734 4 IF .SAVED_CTG THEN FILE_ERROR(BACKUP$_NONCONTIG, .OUTPUT_FAB);
: 3198 4735 3 END;
: 3199 4736 3
: 3200 4737 3
: 3201 4738 3 ! Establish the file version limit.
: 3202 4739 3 !
: 3203 4740 3 IF
: 3204 4741 3 NOT .OUTPUT_NAM[NAM$V_HIGHVER] AND
: 3205 4742 3 NOT .OUTPUT_NAM[NAM$V_LOWVER] AND
: 3206 4743 3 .OUTPUT_ATTBUF[FAR_VERLIMIT] NEQ 0
: 3207 4744 3 THEN
: 3208 4745 4 BEGIN
: 3209 4746 4 LOCAL
: 3210 4747 4 FIB: BBLOCK[FIB$C_LENGTH], ! FIB
: 3211 4748 4 FIB_DESC: VECTOR[2], ! Descriptor for FIB
: 3212 4749 4 FNA_DESC: VECTOR[2], ! Descriptor for n.t.v
: 3213 4750 4 STATUS, ! Status return
: 3214 4751 4 IOSB: VECTOR[4,WORD]; ! I/O status block
: 3215 4752 4
: 3216 4753 4
: 3217 4754 4 ! Set up the FIB and filename descriptor to modify the file version
: 3218 4755 4 ! limit.
: 3219 4756 4 !
: 3220 4757 4 CH$FILL(0, FIB$C_LENGTH, FIB);
: 3221 4758 4 FIB[FIB$V_NORECORD] = TRUE;
: 3222 4759 4 FIB[FIB$W_DID_NUM] = .OUTPUT_NAM[NAM$W_DID_NUM];
: 3223 4760 4 FIB[FIB$W_DID_SEQ] = .OUTPUT_NAM[NAM$W_DID_SEQ];
: 3224 4761 4 FIB[FIB$W_DID_RVN] = .OUTPUT_NAM[NAM$W_DID_RVN];
: 3225 4762 4 FIB[FIB$W_VERLIMIT] = .OUTPUT_ATTBUF[FAR_VERLIMIT];
: 3226 4763 4 FIB_DESC[0] = FIB$C_LENGTH;
: 3227 4764 4 FIB_DESC[1] = FIB;
: 3228 4765 4 FNA_DESC[0] = .OUTPUT_NAM[NAM$S_RSA] + .OUTPUT_NAM[NAM$B_RSL] - .OUTPUT_NAM[NAM$S_NAME];
: 3229 4766 4 FNA_DESC[1] = .OUTPUT_NAM[NAM$S_NAME];
: 3230 P 4767 4 STATUS = $QIOW(
: 3231 P 4768 4 FUNC=IOS MODIFY,
: 3232 P 4769 4 CHAN=.OUTPUT_CHAN,
: 3233 P 4770 4 IOSB=IOSB,
: 3234 P 4771 4 P1=FIB_DESC,
: 3235 4772 4 P2=FNA_DESC);
: 3236 4773 4 IF .STATUS THEN STATUS = .IOSB[0];
: 3237 4774 4 IF NOT .STATUS
```

RESTORE
V04-000

Restore a save set
CREATE_OR_OPEN_FILE - create or open output fil

B 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 101
(18)

```
: 3238      4775 4      THEN
: 3239      4776 4      FILE_ERROR(BACKUP$_WRITEATTR, .OUTPUT_FAB, .STATUS);
: 3240      4777 3      END;
: 3241      4778 2      END;
: 3242      4779 2
: 3243      4780 2
: 3244      4781 2 OUTPUT_FLAGS[OUTPUT_OPEN] = TRUE;
: 3245      4782 2 COM_FLAGS[COM_FILESEEN] = TRUE;
: 3246      4783 2
: 3247      4784 2
: 3248      4785 2 IF .QUAL[QUAL_COMP]
: 3249      4786 2 THEN
: 3250      4787 2 CHECK_EOF_BLOCK();
: 3251      4788 1 END;
```

```
53 2E 41 54 4F 55 51 5D 30 30 30 30 30 30 5B 01649 P.AAP: .ASCII \[000000]QUOTA.SYS;1\
53 2E 41 54 4F 55 51 5D 30 30 30 30 30 30 5B 01658 P.AAQ: .ASCII \[000000]QUOTA.SYS;1\
31 3B 53 59 0165C
31 3B 53 59 0166B
```

.EXTRN SYSS\$OPEN, SYSS\$CREATE

OFFC 00000 CREATE_OR_OPEN_FILE:

```
      5B 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 4263
      5A 00000000G 00 9E 00009 MOVAB STA Q10W, R11
      59 00000000' EF 9E 00010 MOVAB FILE_ERROR, R10
      5E FF10 CE 9E 00017 MOVAB OUTPUT_FAB, R9
      18 A9 14 A9 01 D0 0001C MOVAB -240(SP), SP
      3C A9 10 9C 00020 MOVAB #1, OUTPUT_BLOCK : 4296
      40 A9 B5 00026 ROTL #16, OUTPUT_RECATTR+8, OUTPUT_MAXBLOCK : 4297
      03 12 00029 TSTW OUTPUT_RECATTR+12 : 4298
      18 A9 D7 0002B BNEQ 1$
      0080 C9 18 A9 D1 0002E 1$: DECL OUTPUT_MAXBLOCK : 4299
      F0 8F 34 A9 93 00036 CMPL OUTPUT_MAXBLOCK, OUTPUT_ATTBUF+92 : 4301
      06 13 0003B BGTRU 2$
      00AC C9 18 A9 0080 C9 D0 0003D 2$: MOVAB OUTPUT_ATTBUF+92, OUTPUT_MAXBLOCK : 4304
      03 03 FF16 C9 01 C1 00043 3$: ADDL3 #1, OUTPUT_MAXBLOCK, OUTPUT_ATTBUF+136 : 4305
      009F 31 00050 4$: BBS #3, COM_FLAGS, 5$ : 4308
      FEAB C9 95 00053 5$: BRW 9$
      F7 18 00057 TSTB QUAL+15
      56 04 A9 D0 00059 BGEQ 4$
      50 00C4 C9 D0 0005D MOVAB OUTPUT_NAM, R6 : 4322
      50 10 C0 00062 MOVAB VERIFY_QUAL, R0 : 4323
      03 A6 60 24 A9 81 00065 ADDL2 #16, R0
      04 B6 04 B0 60 28 0006B ADDB3 OUTPUT_FILENAME, (R0), 3(R6) : 4324
      63 28 B9 24 A9 28 00071 MOVAB (R0), 34(R0), 24(R6) : 4331
      56 DD 00077 MOVAB OUTPUT_FILENAME, @OUTPUT_FILENAME+4, (R3) : 4328
      00 00000000G 00 01 FB 00079 PUSHL R6 : 4332
      00 6E 00 00 2C 00080 CALLS #1, INIT_NAMEBLOCK
      C0 AD 00087 MOVAB #0, (SP), #0, #64, FIB : 4338
      0F C0 AD 01 D0 00089 MOVAB #1, FIB : 4339
      FEA6 C9 02 E0 0008D BBS #2, QUAL+10, 6$ : 4341
```

			13	24	A9	B1	00093	CMPW	OUTPUT_FILENAME, #19	4342		
					11	12	00097	BNEQ	7\$	4343		
28	B9	FF3C	CF		13	29	00099	CMPC3	#19, P.AAP, @OUTPUT_FILENAME+4	4349		
					08	12	000A0	BNEQ	7\$	4350		
		C0	AD	00100000	8F	D0	000A2	6\$:	MOVL	#1048576, FIB	4351	
		C2	AD		20	88	000AA	7\$:	BISB2	#32, FIB+2	4353	
		C4	AD		74	A9	D0	000AE	MOVL	OUTPUT_ATTBUF+80, FIB+4	4354	
		C8	AD		78	A9	B0	000B3	MOVW	OUTPUT_ATTBUF+84, FIB+8	4355	
		B8	AD		40	8F	9A	000B8	MOVZBL	#64, FIB_DESC	4360	
		BC	AD		C0	AD	9E	000BD	MOVAB	FIB, FIB_DESC+4		
					7E	7C	000C2	CLRQ	-(SP)			
					7E	7C	000C4	CLRQ	-(SP)			
					7E	D4	000C6	CLRL	-(SP)			
				B8	AD	9F	000C8	PUSHAB	FIB_DESC			
					7E	7C	000CB	CLRQ	-(SP)			
				B0	AD	9F	000CD	PUSHAB	IOSB			
			7E	72	8F	9A	000D0	MOVZBL	#114, -(SP)			
				F8	A9	DD	000D4	PUSHL	OUTPUT_CHAN			
					7E	D4	000D7	CLRL	-(SP)			
		00000000G	00		0C	FB	000D9	CALLS	#12, SYSSQIOW			
			0A		50	E9	000E0	BLBC	STATUS, 8\$	4361		
			50		B0	AD	3C	000E3	MOVZWL	IOSB, STATUS	4362	
			03		50	E9	000E7	BLBC	STATUS, 8\$	4368		
					048C	31	000EA	BRW	50\$	4367		
					50	DD	000ED	8\$:	PUSHL	STATUS	4372	
					00FA	31	000EF	BRW	18\$	4386		
					03	E0	000F2	9\$:	BBS	#3, QUAL+10, 10\$	4391	
					0266	31	000F8	10\$:	BRW	32\$	4393	
					A9	D0	000FB	MOVL	OUTPUT_NAM, R6	4394		
03	A6	OB	A6	04	A9	81	000FF	ADDB3	OUTPUT_FILENAME, 11(R6), 3(R6)	4399		
			50	08	A6	9A	00106	MOVZBL	11(R6), R0	4401		
04	B6	CC	B6		50	28	0010A	MOVZBL	R0, @12(R6), @4(R6)	4406		
	63	28	B9	24	A9	28	00110	MOVZBL	OUTPUT_FILENAME, @OUTPUT_FILENAME+4, (R3)	4407		
					56	DD	00116	MOVZBL	R6	4409		
		00000000G	00		01	FB	00118	PUSHL	R6	4411		
			50		04	A9	D0	0011F	CALLS	#1, INIT_NAMEBLOCK	4413	
			24		74	A9	D0	00123	MOVL	OUTPUT_NAM, R0	4414	
			28		78	A9	B0	00128	MOVL	OUTPUT_ATTBUF+80, 36(R0)	4415	
0040	8F		6E		00	2C	0012D	MOVW	OUTPUT_ATTBUF+84, 40(R0)	4418		
					C0	AD	00134	MOVCS	#0, (SP), #0, #64, FIB	4424		
					74	A9	D0	00136	MOVL	OUTPUT_ATTBUF+80, FIB+4	4426	
					78	A9	B0	0013B	MOVW	OUTPUT_ATTBUF+84, FIB+8	4427	
					40	8F	9A	00140	MOVZBL	#64, FIB_DESC		
					C0	AD	9E	00145	MOVAB	FIB, FIB_DESC+4		
					C4	AD	B5	0014A	TSTW	FIB+4		
					14	12	0014D	BNEQ	11\$			
					C8	AD	B5	0014F	TSTW	FIB+8		
					0F	12	00152	BNEQ	11\$			
					7E	D4	00154	CLRL	-(SP)			
					8F	DD	00156	PUSHL	#BACKUP\$ INVFIID			
		00000000G	00		02	FB	0015C	CALLS	#2, LIB\$SIGNAL			
					C9	95	00163	TSTB	QUAL+8	4418		
					03	19	00167	BLSS	12\$			
					008B	31	00169	BRW	19\$			
					8F	D0	0016C	12\$:	MOVL	#2097152, FIB	4424	
					02	E0	00174	BBS	#2, QUAL+10, 13\$	4426		
OF	FEA6		C9	24	A9	B1	0017A	CMPW	OUTPUT_FILENAME, #19	4427		

28	B9	FE68	CF	11	12	0017E	BNEQ	14\$		
				13	29	00180	CMPC3	#19, P.AAQ, @OUTPUT_FILENAME+4	4428	
				08	12	00187	BNEQ	14\$		
		C0	AD	8F	D0	00189	13\$:	MOVL	#1048576, FIB	4434
		C2	AD	20	88	00191	14\$:	BISB2	#32, FIB+2	4435
				A9	D0	00195		MOVL	OUTPUT_CHAN, R0	4440
		0001FFFF	50	F8	D1	00199		CMPL	R0, #131071	
			8F	1B	1F	001A0		BLSSU	15\$	
				7E	7C	001A2		CLRQ	-(SP)	
				7E	7C	001A4		CLRQ	-(SP)	
				7E	D4	001A6		CLRL	-(SP)	
			B8	AD	9F	001AB		PUSHAB	FIB_DESC	
				7E	7C	001AB		CLRQ	-(SP)	
			B0	AD	9F	001AD		PUSHAB	IOSB	
			7E	72	8F	9A	001B0	MOVZBL	#114, -(SP)	
				50	DD	001B4		PUSHL	R0	
				7E	D4	001B6		CLRL	-(SP)	
			6B	0C	FB	001B8		CALLS	#12, STA_QIOW	
				1D	11	001BB		BRB	16\$	
				7E	7C	001BD	15\$:	CLRQ	-(SP)	
				7E	7C	001BF		CLRQ	-(SP)	
				7E	D4	001C1		CLRL	-(SP)	
			B8	AD	9F	001C3		PUSHAB	FIB_DESC	
				7E	7C	001C6		CLRQ	-(SP)	
			B0	AD	9F	001C8		PUSHAB	IOSB	
			7E	72	8F	9A	001CB	MOVZBL	#114, -(SP)	
				50	DD	001CF		PUSHL	R0	
				7E	D4	001D1		CLRL	-(SP)	
		00000000G	9F	0C	FB	001D3		CALLS	#12, @#SYSSQIOW	
			52	50	D0	001DA	16\$:	MOVL	R0, STATUS	4441
			0A	52	E9	001DD		BLBC	STATUS, 17\$	
			52	B0	AD	3C	001E0	MOVZWL	IOSB, STATUS	4442
			03	52	E9	001E4		BLBC	STATUS, 17\$	
				038F	31	001E7		BRW	50\$	
				52	DD	001EA	17\$:	PUSHL	STATUS	4448
				69	DD	001EC	18\$:	PUSHL	OUTPUT_FAB	4447
				8F	DD	001EE		PUSHL	#BACKUP\$_OPENIN+2	4446
				0157	31	001F4		BRW	30\$	
			2C	AE	9F	001F7	19\$:	PUSHAB	ATR_LIST	4458
		EE05	CF	01	FB	001FA		CALLS	#1, GET_ACP_ATTRIBUTES	
				79	A9	95	001FF	TSTB	OUTPUT_ATTBUF+85	4460
				11	12	00202		BNEQ	20\$	
			03	74	A9	B1	00204	CMPL	OUTPUT_ATTBUF+80, #3	4461
				10	1B	00208		BLEQU	21\$	
			02	0091	C9	91	0020A	CMPL	OUTPUT_ATTBUF+109, #2	4462
				04	12	0020F		BNEQ	20\$	
			09	74	A9	B1	00211	CMPL	OUTPUT_ATTBUF+80, #9	4463
				03	13	00215	20\$:	BEQL	21\$	
				0080	31	00217		BRW	25\$	
				8F	D0	0021A	21\$:	MOVL	#262176, FAT_ATR_LIST	4475
		04	AE	0C	AE	9E	00221	MOVAB	OLD_RECATTR, FAT_ATR_LIST+4	4477
				08	AE	D4	00226	CLRL	FAT_ATR_LIST+8	4478
				7E	D4	00229		CLRL	-(SP)	4484
			04	AE	9F	0022B		PUSHAB	FAT_ATR_LIST	
				7E	7C	0022E		CLRQ	-(SP)	
				7E	D4	00230		CLRL	-(SP)	
			B8	AD	9F	00232		PUSHAB	FIB_DESC	

			7E	7C	00235	CLRQ	-(SP)		
		B0	AD	9F	00237	PUSHAB	I0SB		
			32	DD	0023A	PUSHL	#50		
		F8	A9	DD	0023C	PUSHL	OUTPUT_CHAN		
			7E	D4	0023F	CLRL	-(SP)		
	6B		0C	FB	00241	CALLS	#12, STA QIOW		
	52		50	DD	00244	MOVL	R0, STAT0S		
	07		52	E9	00247	BLBC	STATUS, 22\$	4485	
	52	B0	AD	3C	0024A	MOVZWL	I0SB, STATUS		
	0D		52	E8	0024E	BLBS	STATUS, 23\$	4486	
			52	DD	00251	PUSHL	STATUS	4491	
			69	DD	00253	PUSHL	OUTPUT_FAB	4490	
		00000000G	8F	DD	00255	PUSHL	#BACKUP\$_READATTR	4488	
	6A		03	FB	0025B	CALLS	#3, FILE-ERROR		
38	A9		10	AE	7D	0025E	23\$: MOVQ	OLD RECATR+4, OUTPUT_RECATR+4	4496
			24	A9	9F	00263	PUSHAB	OUTPUT_ATTBUF	4508
			30	AE	9F	00266	PUSHAB	ATR_LIST	
			7E	7C	00269	CLRQ	-(SP)		
			7E	D4	0026B	CLRL	-(SP)		
		B8	AD	9F	0026D	PUSHAB	FIB_DESC		
			7E	7C	00270	CLRQ	-(SP)		
		B0	AD	9F	00272	PUSHAB	I0SB		
			36	DD	00275	PUSHL	#54		
		F8	A9	DD	00277	PUSHL	OUTPUT_CHAN		
			7E	D4	0027A	CLRL	-(SP)		
	6B		0C	FB	0027C	CALLS	#12, STA QIOW		
	52		50	DD	0027F	MOVL	R0, STAT0S		
	08		52	E9	00282	BLBC	STATUS, 24\$	4509	
	52	B0	AD	3C	00285	MOVZWL	I0SB, STATUS		
	01		52	E9	00289	BLBC	STATUS, 24\$	4510	
				04	0028C	RET			
			52	DD	0028D	24\$: PUSHL	STATUS	4515	
			69	DD	0028F	PUSHL	OUTPUT_FAB	4514	
		00000000G	8F	DD	00291	PUSHL	#BACKUP\$_WRITEATTR	4512	
			00B4	31	00297	BRW	30\$		
	50		04	A9	DD	0029A	25\$: MOVL	OUTPUT_NAM, R0	4530
	51		03	A0	9A	0029E	MOVZBL	3(R0), R1	
	51		04	A0	C0	002A2	ADDL2	4(R0), R1	
	51		4C	A0	C3	002A6	SUBL3	76(R0), R1, NAM_DESC	
	28	AE	4C	A0	DD	002AC	MOVL	76(R0), NAM_DESC+4	4531
	CA	AD	7A	A9	DD	002B1	MOVL	OUTPUT_ATTBUF+86, FIB+10	4536
	CE	AD	7E	A9	DD	002B6	MOVW	OUTPUT_ATTBUF+90, FIB+14	4538
	D6	AD	80	8F	88	002BB	BISB2	#128, FIB+22	4539
		0088	C9	95	002C0	TSTB	OUTPUT_FILECHAR	4540	
			04	18	002C4	BGEQ	26\$		
	D6	AD	01	88	002C6	BISB2	#1, FIB+22		
	D8	AD	0080	C9	DD	002CA	26\$: MOVL	OUTPUT_ATTBUF+92, FIB+24	4541
			24	A9	9F	002D0	PUSHAB	OUTPUT_ATTBUF	4553
			30	AE	9F	002D3	PUSHAB	ATR_LIST	
			7E	7C	002D6	CLRQ	-(SP)		
		34	AE	9F	002D8	PUSHAB	NAM_DESC		
		B8	AD	9F	002DB	PUSHAB	FIB_DESC		
			7E	7C	002DE	CLRQ	-(SP)		
		B0	AD	9F	002E0	PUSHAB	I0SB		
	7E		F3	8F	9A	002E3	MOVZBL	#243, -(SP)	
			F8	A9	DD	002E7	PUSHL	OUTPUT_CHAN	
			7E	D4	002EA	CLRL	-(SP)		

	6B		0C	FB	002EC	CALLS	#12, STA QIOW		
	52		50	DO	002EF	MOVL	R0, STATUS		
	04		52	E9	002F2	BLBC	STATUS, 27\$		4554
	52	B0	AD	3C	002F5	MOVZWL	IOSB, STATUS		
00000850	8F		52	D1	002F9	CMPL	STATUS, #2128		4557
			3F	12	00300	BNEQ	28\$		
	3B	D6	AD	E9	00302	BLBC	FIB+22, 28\$		
D6	AD		01	8A	00306	BICB2	#1, FIB+22		4563
		24	A9	9F	0030A	PUSHAB	OUTPUT_ATTBUF		4571
		30	AE	9F	0030D	PUSHAB	ATR_LIST		
			7E	7C	00310	CLRQ	-(SP)		
		34	AE	9F	00312	PUSHAB	NAM_DESC		
		B8	AD	9F	00315	PUSHAB	FIB_DESC		
			7E	7C	00318	CLRQ	-(SP)		
		B0	AD	9F	0031A	PUSHAB	IOSB		
	7E	F3	8F	9A	0031D	MOVZBL	#243, -(SP)		
		F8	A9	DD	00321	PUSHL	OUTPUT_CHAN		
			7E	D4	00324	CLRL	-(SP)		
	6B		0C	FB	00326	CALLS	#12, STA QIOW		
	52		50	DO	00329	MOVL	R0, STATUS		
	15		52	E9	0032C	BLBC	STATUS, 29\$		4572
	52	B0	AD	3C	0032F	MOVZWL	IOSB, STATUS		
	0E		52	E9	00333	BLBC	STATUS, 29\$		4573
			69	DD	00336	PUSHL	OUTPUT_FAB		
		00000000G	8F	DD	00338	PUSHL	#BACKUP\$NONCONTIG		
	6A		02	FB	0033E	CALLS	#2, FILE_ERROR		
	0E		52	E8	00341	BLBS	STATUS, 31\$		4577
			52	DD	00344	PUSHL	STATUS		4583
			69	DD	00346	PUSHL	OUTPUT_FAB		4582
		00000000G	8F	DD	00348	PUSHL	#BACKUP\$OPENOUT+2		4581
	6A		03	FB	0034E	CALLS	#3, FILE_ERROR		
			04	04	00351	RET			4579
		008C	C9	D5	00352	TSTL	OUTPUT_ATTBUF+104		4591
			38	13	00356	BEQL	34\$		
00000000G	00		00	FB	00358	CALLS	#0, STA_WRITEBOOT		4593
			2F	11	0035F	BRB	34\$		4372
	52		69	DO	00361	MOVL	OUTPUT_FAB, R2		4600
		FEA4	C9	95	00364	TSTB	QUAL+8		4597
			29	18	00368	BGEQ	35\$		
			52	DD	0036A	PUSHL	R2		4600
00000000G	00		01	FB	0036C	CALLS	#1, SYS\$OPEN		
	12		50	E8	00373	BLBS	R0, 33\$		
	50		69	DO	00376	MOVL	OUTPUT_FAB, R0		4606
	7E	08	A0	7D	00379	MOVQ	8(R0), -(SP)		
			50	DD	0037D	PUSHL	R0		4605
		00000000G	8F	DD	0037F	PUSHL	#BACKUP\$OPENIN+2		4604
			00E9	31	00385	BRW	44\$		
	50		69	DO	00388	MOVL	OUTPUT_FAB, R0		4609
	F8	A9	0C	A0	0038B	MOVL	12(R0), OUTPUT_CHAN		
			01E6	31	00390	BRW	50\$		4597
0001C04A	8F	08	A2	D1	00393	CMPL	8(R2), #114762		4622
			08	12	0039B	BNEQ	36\$		
	F4F7	CF	00	FB	0039D	CALLS	#0, IMPLICIT_DIRECTORY		4624
		75	50	E9	003A2	BLBC	R0, 40\$		
0D	FEAB	C9	02	E0	003A5	BBS	#2, QUAL+12, 37\$		4631
07	FEA6	C9	04	E0	003AB	BBS	#4, QUAL+10, 37\$		
	0084	C9	FD70	C9	DO	003B1	MOVL	JPI_UIC, OUTPUT_ATTBUF+96	4633

RESTORE
V04-000

Restore a save set
CREATE_OR_OPEN_FILE - create or open output fil

G 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 106
(18)

		07	FEA8	C9		04	E1	003B8	37\$:	BBC	#4, QUAL+12, 38\$	4635
			0084	C9	FEDC	C9	DO	003BE		MOVL	QUAL+64, OUTPUT_ATTBUF+96	4637
		0A	FEA8	C9		03	E1	003C5	38\$:	BBC	#3, QUAL+12, 39\$	4639
			ED16	CF		00	FB	003CB		CALLS	#0, GET_PARENT_UIC	4641
0058	8F		0084	C9		50	DO	003D0		MOVL	R0, OUTPUT_ATTBUF+96	
		00		6E		00	2C	003D5	39\$:	MOVC5	#0, (SP), #0, #88, \$RMS_PTR	4646
					A8	AD		003DC				
			A8	AD	5813	8F	BO	003DE		MOVW	#22547, \$RMS_PTR	
			B4	AD	0084	C9	DO	003E4		MOVL	OUTPUT_ATTBUF+96, XABPRO+12	4647
			BO	AD		01	AE	003EA		MNEGW	#1, XABPRO+8	4648
				50		69	DO	003EE		MOVL	OUTPUT_FAB, R0	4649
			24	A0	A8	AD	9E	003F1		MOVAB	XABPRO, 36(R0)	
				57	10	A0	DO	003F6		MOVL	16(R0), SAVED_ALQ	4654
58	06	A0		01		04	EF	003FA		EXTZV	#4, #1, 6(R0), SAVED_CTG	4655
						50	DD	00400		PUSHL	R0	4656
		00000000G		00		01	FB	00402		CALLS	#1, SYSS\$CREATE	
				52		50	DO	00409		MOVL	R0, STATUS	
		00018292		8F		52	D1	0040C		CMPL	STATUS, #98962	4662
						15	12	00413		BNEQ	42\$	
			F47F	CF		00	FB	00415		CALLS	#0, IMPLICIT_DIRECTORY	4665
				01		50	E8	0041A	40\$:	BLBS	R0, 41\$	
						04		0041D		RET		
						69	DD	0041E	41\$:	PUSHL	OUTPUT_FAB	4666
		00000000G		00		01	FB	00420		CALLS	#1, SYSS\$CREATE	
				52		50	DO	00427		MOVL	R0, STATUS	
		00018544		8F		52	D1	0042A	42\$:	CMPL	STATUS, #99652	4670
						29	12	00431		BNEQ	43\$	
				50		69	DO	00433		MOVL	OUTPUT_FAB, R0	
21	06			A0		04	E1	00436		BBC	#4, 6(R0), 43\$	
				50		69	DO	0043B		MOVL	OUTPUT_FAB, R0	4676
	06			A0		10	8A	0043E		BICB2	#16, 6(R0)	
						50	DD	00442		PUSHL	R0	4677
		00000000G		00		01	FB	00444		CALLS	#1, SYSS\$CREATE	
				52		50	DO	0044B		MOVL	R0, STATUS	
				0B		52	E9	0044E		BLBC	STATUS, 43\$	4678
						69	DD	00451		PUSHL	OUTPUT_FAB	
					00000000G	8F	DD	00453		PUSHL	#BACKUP\$_NONCONTIG	
			6A			02	FB	00459		CALLS	#2, FILE_ERROR	
			50			69	DO	0045C	43\$:	MOVL	OUTPUT_FAB, R0	4682
					24	A0	D4	0045F		CLRL	36(R0)	
				10		52	E8	00462		BLBS	STATUS, 45\$	4685
				7E	08	A0	7D	00465		MOVQ	8(R0), -(SP)	4691
						50	DD	00469		PUSHL	R0	4690
					00000000G	8F	DD	0046B		PUSHL	#BACKUP\$_OPENOUT+2	4689
			6A			04	FB	00471	44\$:	CALLS	#4, FILE_ERROR	
						04		00474		RET		4687
				56		69	DO	00475	45\$:	MOVL	OUTPUT_FAB, R6	4696
			FB	A9	0C	A6	DO	00478		MOVL	12(R6), OUTPUT_CHAN	
74		FEA7		C9		04	E1	0047D		BBC	#4, QUAL+11, 48\$	4702
				57	10	A6	D1	00483		CMPL	16(R6), SAVED_ALQ	
						6E	1E	00487		BGEQU	48\$	
0040	8F			6E		00	2C	00489		MOVC5	#0, (SP), #0, #64, FIB	4712
					58	AE		00490				
			5A	AE		20	88	00492		BISB2	#32, FIB+2	4713
			6E	AE	80	8F	88	00496		BISB2	#128, FIB+22	4714
70	AE			57	10	A6	C3	0049B		SUBL3	16(R6), SAVED_ALQ, FIB+24	4715
			50	AE	40	8F	9A	004A1		MOVZBL	#64, FIB_DESC	4716

RESTORE
V04-000

Restore a save set
CREATE_OR_OPEN_FILE - create or open output fil

H 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 107
(18)

54	AE	58	AE	9E	004A6	MOVAB	FIB, FIB_DESC+4	4717
			7E	7C	004AB	CLRQ	-(SP)	4722
			7E	7C	004AD	CLRQ	-(SP)	
			7E	D4	004AF	CLRL	-(SP)	
		64	AE	9F	004B1	PUSHAB	FIB_DESC	
			7E	7C	004B4	CLRQ	-(SP)	
		68	AE	9F	004B6	PUSHAB	IOSB	
			36	DD	004B9	PUSHL	#54	
		F8	A9	DD	004BB	PUSHL	OUTPUT_CHAN	
			7E	D4	004BE	CLRL	-(SP)	
00000000G	00		0C	FB	004C0	CALLS	#12, SYSSQIOW	
	07		50	E9	004C7	BLBC	STATUS, 46\$	4723
	50	48	AE	3C	004CA	MOVZWL	IOSB, STATUS	
	18		50	E8	004CE	BLBS	STATUS, 47\$	4724
			50	DD	004D1	PUSHL	STATUS	4730
			69	DD	004D3	PUSHL	OUTPUT_FAB	4729
		00000000G	8F	DD	004D5	PUSHL	#BACKUP\$ OPENOUT+2	4728
	6A		03	FB	004DB	CALLS	#3, FILE_ERROR	
		F8	A9	DD	004DE	PUSHL	OUTPUT_CHAN	4731
00000000G	00		01	FB	004E1	CALLS	#1, SYSSDASSGN	
			04	04	004E8	RET		4726
	0B		58	E9	004E9	BLBC	SAVED CTG, 48\$	4734
			69	DD	004EC	PUSHL	OUTPUT_FAB	
		00000000G	8F	DD	004EE	PUSHL	#BACKUP\$ NONCONTIG	
	6A		02	FB	004F4	CALLS	#2, FILE_ERROR	
	56	04	A9	DD	004F7	MOVL	OUTPUT_NAM, R6	4741
		35	A6	95	004FB	TSTB	53(R6)	
			79	19	004FE	BLSS	50\$	
74	35	A6	06	E0	00500	BBS	#6, 53(R6), 50\$	4742
		00A2	C9	B5	00505	TSTW	OUTPUT_ATTBUF+126	4743
			6E	13	00509	BEQL	50\$	
0040	8F	00	00	2C	0050B	MOVCS	#0, (SP), #0, #64, FIB	4757
		58	AE		00512			
	5A	AE	20	88	00514	BISB2	#32, FIB+2	4758
	62	AE	A6	DD	00518	MOVL	42(R6), FIB+10	4759
	66	AE	2E	A6	B0	MOVW	46(R6), FIB+14	4761
	94	AD	00A2	C9	B0	MOVW	OUTPUT_ATTBUF+126, FIB+44	4762
	50	AE	40	8F	9A	MOVZBL	#64, FIB_DESC	4763
	54	AE	58	AE	9E	MOVAB	FIB, FIB_DESC+4	4764
		50	03	A6	9A	MOVZBL	3(R6), R0	4765
		50	04	A6	C0	ADDL2	4(R6), R0	
48	AE		4C	A6	C3	SUBL3	76(R6), R0, FNA_DESC	
		4C	AE	A6	DD	MOVL	76(R6), FNA_DESC+4	4766
				7E	7C	CLRQ	-(SP)	4772
				7E	7C	CLRQ	-(SP)	
		58	AE	9F	00549	PUSHAB	FNA_DESC	
		64	AE	9F	0054C	PUSHAB	FIB_DESC	
			7E	7C	0054F	CLRQ	-(SP)	
		60	AE	9F	00551	PUSHAB	IOSB	
			36	DD	00554	PUSHL	#54	
		F8	A9	DD	00556	PUSHL	OUTPUT_CHAN	
			7E	D4	00559	CLRL	-(SP)	
00000000G	00		0C	FB	0055B	CALLS	#12, SYSSQIOW	
	07		50	E9	00562	BLBC	STATUS, 49\$	4773
	50	40	AE	3C	00565	MOVZWL	IOSB, STATUS	
	0D		50	E8	00569	BLBS	STATUS, 50\$	4774
			50	DD	0056C	PUSHL	STATUS	4776

RESTORE
V04-000

Restore a save set
CREATE_OR_OPEN_FILE - create or open output fil

1 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 108
(18)

		00000000G	69	DD	0056E	PUSHL	OUTPUT_FAB	:	
			8F	DD	00570	PUSHL	#BACKUP\$_WRITEATTR	:	
	6A		03	FB	00576	CALLS	#3, FILE_ERROR	:	
FC	A9		01	88	00579	50\$:	BISB2	#1, OUTPUT_FLAGS	4781
FF16	C9		04	88	0057D		BISB2	#4, COM_FLAGS	4782
		FEA4	C9	95	00582		TSTB	QUAL+8	4785
			05	18	00586		BGEQ	51\$	:
F07E	CF		00	FB	00588		CALLS	#0, CHECK_EOF_BLOCK	4787
			04	0058D	51\$:	RET			4788

; Routine Size: 1422 bytes, Routine Base: CODE + 166F

RESTORE
V04-000

Restore a save set
RESTORE_ERROR - handle write errors

J 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 109
(19)

```

3253 4789 1 %SBTTL 'RESTORE_ERROR - handle write errors'
3254 4790 1 ROUTINE RESTORE_ERROR(BCB): NOVALUE=
3255 4791 1
3256 4792 1 ++
3257 4793 1
3258 4794 1 FUNCTIONAL DESCRIPTION:
3259 4795 1 This is the error action routine for asynchronous writes.
3260 4796 1
3261 4797 1 INPUT PARAMETERS:
3262 4798 1 BCB - BCB on which the error occurred.
3263 4799 1
3264 4800 1 IMPLICIT INPUTS:
3265 4801 1 NONE
3266 4802 1
3267 4803 1 OUTPUT PARAMETERS:
3268 4804 1 NONE
3269 4805 1
3270 4806 1 IMPLICIT OUTPUTS:
3271 4807 1 NONE
3272 4808 1
3273 4809 1 ROUTINE VALUE:
3274 4810 1 NONE
3275 4811 1
3276 4812 1 SIDE EFFECTS:
3277 4813 1 NONE
3278 4814 1
3279 4815 1 --
3280 4816 1
3281 4817 2 BEGIN
3282 4818 2 MAP
3283 4819 2 BCB: REF BBLOCK; ! Pointer to BCB
3284 4820 2 LOCAL
3285 4821 2 REC: REF BBLOCK, ! Pointer to record
3286 4822 2 STATUS, ! Status variable
3287 4823 2 IOSB: VECTOR[4,WORD]; ! I/O status block
3288 4824 2
3289 4825 2
3290 4826 2 ! Point to the record on which the error occurred.
3291 4827 2
3292 4828 2 REC = .BCB[BCB_RECORD];
3293 4829 2
3294 4830 2
3295 4831 2 ! Loop for all blocks in the record reissuing the writes and reporting
3296 4832 2 any errors that occur.
3297 4833 2
3298 4834 2 INCR XBLOCK FROM 0 TO .REC[BRH$W_RSIZE]^9 - 1 DO
3299 4835 2 BEGIN
3300 4836 3 STATUS = CSQIOW(
3301 4837 3 FUNC=.OUTPUT_FUNC,
3302 4838 3 CHAN=.OUTPUT_CHAN,
3303 4839 3 IOSB=IOSB,
3304 4840 3 P1=.REC + BRH$C_LENGTH + .XBLOCK^9,
3305 4841 3 P2=512,
3306 4842 3 P3=.REC[BRH$L_ADDRESS] + .XBLOCK);
3307 4843 3 IF .STATUS THEN STATUS = .IOSB[0];
3308 4844 3 IF NOT .STATUS THEN IF NOT FIND_BADBLOCK(.O)PUT_BAD, .REC[BRH$L_ADDRESS] + .XBLOCK)
3309 4845 3 THEN
```

RESTORE
V04-000

Restore a save set
RESTORE_ERROR - handle write errors

K 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 110
(19)

```
: 3310      4846 3      FILE ERROR(  
: 3311      4847 3      BACKUP$ WRITEBLOCK,  
: 3312      4848 3      .OUTPUT_FAB,  
: 3313      4849 3      .REC[BR$SL_ADDRESS] + .XBLOCK,  
: 3314      4850 3      .STATUS);  
: 3315      4851 2      END;  
: 3316      4852 1      END;
```

				00FC 000C0 RESTORE_ERROR:		
				.WORD	Save R2,R3,R4,R5,R6,R7	4790
				MOVAB	OUTPUT_FUNC, R7	
				SUBL2	#8, SP	
				MOVL	BCB, R0	4828
				MOVL	16(R0), REC	
				MOVZWL	(REC), R6	4834
				ASHL	#-9, R6, R6	
				MNEGL	#1, XBLOCK	4842
				BRW	5\$	
				ASHL	#9, XBLOCK, R0	
				MOVAB	16(R0)[REC], R1	
				MOVL	OUTPUT_CHAN, R0	
				CML	R0, #131071	
				BLSSU	2\$	
				CLRQ	-(SP)	
				CLRL	-(SP)	
				ADDL3	8(REC), XBLOCK, R4	
				PUSHL	R4	
				MOVZWL	#512, -(SP)	
				PUSHL	R1	
				CLRQ	-(SP)	
				PUSHAB	IOSB	
				MOVZBL	OUTPUT_FUNC, -(SP)	
				PUSHL	R0	
				CLRL	-(SP)	
				CALLS	#12, STA_QIOW	
				BRB	3\$	
				CLRQ	-(SP)	
				CLRL	-(SP)	
				ADDL3	8(REC), XBLOCK, R4	
				PUSHL	R4	
				MOVZWL	#512, -(SP)	
				PUSHL	R1	
				CLRQ	-(SP)	
				PUSHAB	IOSB	
				MOVZBL	OUTPUT_FUNC, -(SP)	
				PUSHL	R0	
				CLRL	-(SP)	
				CALLS	#12, @#SYSSQIOW	
				MOVL	R0, STATUS	
				BLBC	STATUS, 4\$	4843
				MOVZWL	IOSB, STATUS	
				BLBS	STATUS, 5\$	4844
				PUSHL	R4	

RESTORE
V04-000

Restore a save set
RESTORE_ERROR - handle write errors

L 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 111
(19)

00000000G	00	00B2	C7	DD	00093	PUSHL	OUTPUT BAD	:
	13		02	FB	00097	CALLS	#2, FIND_BADBLOCK	:
			50	E8	0009E	BLBS	R0, 5\$	:
			30	BB	000A1	PUSHR	#*M<R4,R5>	: 4849
		00A2	C7	DD	000A3	PUSHL	OUTPUT FAB	: 4848
		00000000G	8F	DD	000A7	PUSHL	#BACKUP\$ WRITEBLOCK	: 4846
01	00000000G	00	04	FB	000AD	CALLS	#4, FILE_ERROR	:
	53		56	F2	000B4	AOBLSS	R6, XBLOCK, 6\$	: 4834
				04	000B8	RET		: 4852
			FF66	31	000B9	BRW	1\$	: 4834
					6\$:			:

; Routine Size: 188 bytes, Routine Base: CODE + 1BFD

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

M 12
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 112
(20)

```
3318 4853 1 %SBTTL 'RESTORE ONE RECORD - restore save set record'
3319 4854 1 GLOBAL ROUTINE RESTORE_ONE_RECORD(REC,BCB): NOVALUE=
3320 4855 1
3321 4856 1 ++
3322 4857 1
3323 4858 1 FUNCTIONAL DESCRIPTION:
3324 4859 1 This routine restores the contents of one save set record.
3325 4860 1
3326 4861 1 INPUT PARAMETERS:
3327 4862 1 REC - Pointer to record.
3328 4863 1 BCB - Pointer to BCB.
3329 4864 1
3330 4865 1 IMPLICIT INPUTS:
3331 4866 1 NONE
3332 4867 1
3333 4868 1 OUTPUT PARAMETERS:
3334 4869 1 NONE
3335 4870 1
3336 4871 1 IMPLICIT OUTPUTS:
3337 4872 1 NONE
3338 4873 1
3339 4874 1 ROUTINE VALUE:
3340 4875 1 NONE
3341 4876 1
3342 4877 1 SIDE EFFECTS:
3343 4878 1 NONE
3344 4879 1
3345 4880 1 --
3346 4881 1
3347 4882 2 BEGIN
3348 4883 2 MAP
3349 4884 2 REC: REF BBLOCK, ! Pointer to record
3350 4885 2 BCB: REF BBLOCK; ! Pointer to BCB
3351 4886 2
3352 4887 2
3353 4888 2 OUTPUT_BCB = .BCB;
3354 4889 2
3355 4890 2
3356 4891 2 CASE .REC[BRH$W_RTYPE] FROM BRH$K_SUMMARY TO BRH$K_FILE_EXT OF
3357 4892 2 SET
3358 4893 2
3359 4894 2
3360 4895 2 [OUTRANGE]:
3361 4896 2 0;
3362 4897 2
3363 4898 2
3364 4899 2 [BRH$K_VBN, BRH$K_LBN]:
3365 4900 3 BEGIN
3366 4901 3
3367 4902 3 ! Test for an unexpected change in file ID. This indicates that some
3368 4903 3 ! save set data has been lost. Close the current output file, if any,
3369 4904 3 ! and, if selected, create the file from which this data came.
3370 4905 3
3371 4906 3 IF CH$NEQ(
3372 4907 3 FID$C_LENGTH, OUTPUT_ATTBUF[FAR FID],
3373 4908 3 FID$C_LENGTH, BBLOCK[BCB[BCB_BOFFER], BBH$W_FID])
3374 4909 3 THEN
```

```

: 3375      4910 4      BEGIN
: 3376      4911 4      LOCAL
: 3377      4912 4      BUFFER:          REF BBLOCK;      ! Pointer to buffer
: 3378      4913 4
: 3379      4914 4
: 3380      4915 4      ! Warn the user that this has occurred. However, if this is a
: 3381      4916 4      ! continuation tape verify pass, it is expected to occur.
: 3382      4917 4
: 3383      4918 4      IF
: 3384      4919 4          NOT .OUTPUT_FLAGS[OUTPUT_V_FIRST] OR
: 3385      4920 4          NOT .COM_FLAGS[COM_VERIFYING] OR
: 3386      4921 4          .BBLOCK[BCB[BCB_BUFFER], BBH$W_VOLNUM] LEQ 1
: 3387      4922 4      THEN
: 3388      4923 4          FILE_ERROR(
: 3389      4924 4              BACKUP$_FIDERROR,
: 3390      4925 4              .OUTPUT_FAB);
: 3391      4926 4
: 3392      4927 4
: 3393      4928 4      ! Close the current output file, if any.
: 3394      4929 4      !
: 3395      4930 4      CLOSE_RESTORE();
: 3396      4931 4
: 3397      4932 4
: 3398      4933 4      ! Avoid situations where we should not restore the file.
: 3399      4934 4      !
: 3400      4935 4      IF
: 3401      4936 4          (.REC[BRH$V_DIRECTORY] AND NOT .QUAL[QUAL_IMAG]) OR
: 3402      4937 4          .QUAL[QUAL_VOLU] OR
: 3403      4938 4          .QUAL[QUAL_INCR]
: 3404      4939 4      THEN
: 3405      4940 4          RETURN;
: 3406      4941 4
: 3407      4942 4
: 3408      4943 4      ! Synthesize the file attribute buffer from the information in the
: 3409      4944 4      ! save set block header.
: 3410      4945 4      !
: 3411      4946 4      BUFFER = .BCB[BCB_BUFFER];
: 3412      4947 4      CH$FILL(0, FAR_LENGTH, OUTPUT_ATTBUF);
: 3413      4948 4      OUTPUT_ATTBUF[FAR_FID_NUM] = .BUFFER[BBH$W_FID_NUM];
: 3414      4949 4      OUTPUT_ATTBUF[FAR_FID_SEQ] = .BUFFER[BBH$W_FID_SEQ];
: 3415      4950 4      OUTPUT_ATTBUF[FAR_FID_RVNW] = .BUFFER[BBH$W_FID_RVN];
: 3416      4951 4      OUTPUT_FILENAME[DSC$W_LENGTH] = (.BUFFER[BBH$T_FILENAME])<0,8>;
: 3417      4952 4      OUTPUT_FILENAME[DSC$A_POINTER] = .BUFFER[BBH$T_FILENAME] + 1;
: 3418      4953 4      OUTPUT_RECATTR[FAT$B_RTYPE] = .BUFFER[BBH$B_RTYPE];
: 3419      4954 4      OUTPUT_RECATTR[FAT$B_RATTRIB] = .BUFFER[BBH$B_RATTRIB];
: 3420      4955 4      OUTPUT_RECATTR[FAT$W_RSIZE] = .BUFFER[BBH$W_RSIZE];
: 3421      4956 4      OUTPUT_RECATTR[FAT$L_EFBLK] = ROT(.BUFFER[BBH$L_FILESIZE] + 1, 16);
: 3422      4957 4      OUTPUT_RECATTR[FAT$B_BKTSIZE] = .BUFFER[BBH$B_BKTSIZE];
: 3423      4958 4      OUTPUT_RECATTR[FAT$B_VFCSIZE] = .BUFFER[BBH$B_VFCSIZE];
: 3424      4959 4      OUTPUT_RECATTR[FAT$W_MAXREC] = .BUFFER[BBH$W_MAXREC];
: 3425      4960 4      OUTPUT_ATTBUF[FAR_FI$SIZE] = .BUFFER[BBH$L_FILESIZE];
: 3426      4961 4
: 3427      4962 4
: 3428      4963 4      ! Determine if the new file is selected, and if it is, create
: 3429      4964 4      ! or open it.
: 3430      4965 4      !
: 3431      4966 4      IF INITIALIZE_AND_SELECT()
```

```

: 3432      4967 4      THEN
: 3433      4968 5          BEGIN
: 3434      4969 5              CREATE OR OPEN FILE();
: 3435      4970 5              IF .OUTPUT_FLAGS[OUTPUT_OPEN]
: 3436      4971 5                  THEN OUTPUT_FLAGS[OUTPUT_IMPLICIT] = TRUE;
: 3437      4972 4              END;
: 3438      4973 3          END;
: 3439      4974 3
: 3440      4975 3
: 3441      4976 3      : If special /INCREMENTAL directory processing is enabled, do it.
: 3442      4977 3      IF .OUTPUT_FLAGS[OUTPUT_INCR_DIR]
: 3443      4978 3      THEN
: 3444      4979 3          INCREMENTAL_DIRECTORY(.REC[BRH$W_RSIZ], .REC + BRH$C_LENGTH);
: 3445      4980 3
: 3446      4981 3
: 3447      4982 3
: 3448      4983 3      : If the NONSEQUENTIAL flag is set in the record header then
: 3449      4984 3      : set our internal flag
: 3450      4985 3      IF .REC[BRH$V_NONSEQUENTIAL]
: 3451      4986 3      THEN
: 3452      4987 3          OUTPUT_FLAGS[OUTPUT_NONSEQ_IN] = TRUE ;
: 3453      4988 3
: 3454      4989 3
: 3455      4990 3      : If the file from which this data came is selected, write the
: 3456      4991 3      : data.
: 3457      4992 3      IF .OUTPUT_FLAGS[OUTPUT_OPEN]
: 3458      4993 3      THEN
: 3459      4994 3          BEGIN
: 3460      4995 4              LOCAL
: 3461      4996 4                  IOSB:          VECTOR[4,WORD], ! I/O status block
: 3462      4997 4                  THIS_MINVBN,      ! Smallest block number
: 3463      4998 4                  THIS_MAXVBN,      ! Largest block number
: 3464      4999 4                  THIS_BLOCKS,      ! Blocks in current record
: 3465      5000 4                  STATOS;          ! Status variable
: 3466      5001 4
: 3467      5002 4
: 3468      5003 4
: 3469      5004 4      : Compute the smallest and largest block numbers in this record.
: 3470      5005 4      :
: 3471      5006 4      THIS_MINVBN = .REC[BRH$L_ADDRESS];
: 3472      5007 4      THIS_BLOCKS = .REC[BRH$W_RSIZ] ^ -9;
: 3473      5008 4      THIS_MAXVBN = .THIS_MINVBN + .THIS_BLOCKS - 1;
: 3474      5009 4
: 3475      5010 4
: 3476      5011 4      : Make sure that the block number is at least as large as the
: 3477      5012 4      : next block number we are expecting. However, if the NONSEQUENTIAL
: 3478      5013 4      : flag is set in the record header accept the error. This flag
: 3479      5014 4      : is set by the HSC when it performs it's own backup (PHYSICAL saveset)
: 3480      5015 4      :
: 3481      5016 4      IF (.THIS_MINVBN LSSU .OUTPUT_BLOCK) AND
: 3482      5017 4      NOT .OUTPUT_FLAGS[OUTPUT_NONSEQ_IN]
: 3483      5018 4      THEN
: 3484      5019 5          BEGIN
: 3485      5020 5              FILE_ERROR(
: 3486      5021 5                  BACKUP$_VBNERROR,
: 3487      5022 5                  .OUTPUT_FAB,
: 3488      5023 5                  .THIS_MINVBN, .OUTPUT_BLOCK-1);
```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

C 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 115
(20)

```

3489      5024 4      END;
3490      5025 4
3491      5026 4
3492      5027 4      ! Make sure that no blocks were skipped. However, if this is
3493      5028 4      ! the first data record in a continuation tape verify pass,
3494      5029 4      ! accept the error. Accept the error if /VOLUME is specified.
3495      5030 4      ! If the NONSEQUENTIAL
3496      5031 4      ! flag is set in the record header accept the error. This flag
3497      5032 4      ! is set by the HSC when it performs it's own backup (PHYSICAL saveset)
3498      5033 4
3499      5034 4      IF
3500      5035 4          NOT .QUAL[QUAL_VOLU] AND
3501      5036 4          .THIS_MINVBN GTRU .OUTPUT_BLOCK AND
3502      5037 5          (NOT .OUTPUT_FLAGS[OUTPUT_V_FIRST] OR
3503      5038 5          NOT .COM_FLAGS[COM_VERIFYING] OF
3504      5039 4          .BLOCK[BCB[BCB_BUFFER], BBH$W_VOLNUM] LEQ 1) AND
3505      5040 4          NOT .OUTPUT_FLAGS[OUTPUT_NONSEQ_IN]
3506      5041 4      THEN
3507      5042 5          BEGIN
3508      5043 5              FILE_ERROR(
3509      5044 5                  BACKUP$_VBNMISSING,
3510      5045 5                  .OUTPUT_FAB,
3511      5046 5                  .OUTPUT_BLOCK, .THIS_MINVBN-1);
3512      5047 4          END;
3513      5048 4
3514      5049 4
3515      5050 4      ! Make sure that the upper limit does not exceed the boundaries
3516      5051 4      ! of the file that we were expecting. If it does, do not process
3517      5052 4      ! any blocks beyond the boundary.
3518      5053 4
3519      5054 4      IF .THIS_MAXVBN GTRU .OUTPUT_MAXBLOCK
3520      5055 4      THEN
3521      5056 5          BEGIN
3522      5057 5              FILE_ERROR(
3523      5058 5                  BACKUP$_VBNPASTEOF,
3524      5059 5                  .OUTPUT_FAB,
3525      5060 5                  .OUTPUT_MAXBLOCK+1, .THIS_MAXVBN);
3526      5061 5              THIS_BLOCKS = MAX(0, .OUTPUT_MAXBLOCK - .THIS_MINVBN + 1);
3527      5062 4          END;
3528      5063 4
3529      5064 4
3530      5065 4      ! If the data had an error when it was saved, report the fact.
3531      5066 4      ! Omit the report for a disk-to-disk operation.
3532      5067 4
3533      5068 4      IF .REC[BRH$V_BADDATA] AND NOT .QUAL[QUAL_IF11]
3534      5069 4      THEN
3535      5070 4          FILE_ERROR(
3536      5071 4              BACKUP$_BADDATA,
3537      5072 4              .OUTPUT_FAB,
3538      5073 4              .THIS_MINVBN);
3539      5074 4
3540      5075 4
3541      5076 4      IF .QUAL[QUAL_COMP]
3542      5077 4      THEN
3543      5078 5          BEGIN
3544      5079 5
3545      5080 5              ! for a compare operation, read the corresponding blocks from
```

```
3546      5081 5      ! the output medium into the compare buffer. If an error
3547      5082 5      ! occurs, re-execute the read one block at a time reporting
3548      5083 5      ! the errors.
3549      5084 5
3550      P 5085 5      STATUS = C$QIOW(
3551      P 5086 5      FUNC=.INPUT_FUNC,
3552      P 5087 5      CHAN=.OUTPUT_CHAN,
3553      P 5088 5      IOSB=IOSB,
3554      P 5089 5      P1=.COMPARE_BCB[BCB_BUFFER] + BBH$C_LENGTH,
3555      P 5090 5      P2=.THIS_BLOCKS ^ 9,
3556      5091 5      P3=.THIS_MINVBN);
3557      5092 5      IF .STATUS THEN STATUS = .IOSB[0];
3558      5093 5      IF NOT .STATUS
3559      5094 5      THEN
3560      5095 6          BEGIN
3561      5096 6              INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
3562      5097 7                  BEGIN
3563      P 5098 7                      STATUS = C$QIOW(
3564      P 5099 7                      FUNC=.INPUT_FUNC,
3565      P 5100 7                      CHAN=.OUTPUT_CHAN,
3566      P 5101 7                      IOSB=IOSB,
3567      P 5102 7                      P1=.COMPARE_BCB[BCB_BUFFER] + BBH$C_LENGTH + .XVBN^9,
3568      P 5103 7                      P2=512,
3569      5104 7                      P3=.REC[BRH$L_ADDRESS] + .XVBN);
3570      5105 7                      IF .STATUS THEN STATUS = .IOSB[0];
3571      5106 7                      IF NOT .STATUS THEN IF NOT FIND_BADBLOCK(.OUTPUT_BAD, .REC[BRH$L_ADDRESS] + .XVBN)
3572      5107 7                      THEN
3573      5108 7                          FILE_ERROR(
3574      5109 7                              BACKUP$_READBLOCK,
3575      5110 7                              .OUTPUT_FAB,
3576      5111 7                              .REC[BRH$L_ADDRESS] + .XVBN,
3577      5112 7                              .STATUS);
3578      5113 6                      END;
3579      5114 5                  END;
3580      5115 5
3581      5116 5
3582      5117 5      ! Compare the save set data to the output medium data,
3583      5118 5      ! reporting discrepancies.
3584      5119 5
3585      5120 5      INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
3586      5121 6          BEGIN
3587      5122 6              IF CH$NEQ(
3588      5123 6                  512, .REC + BRH$C_LENGTH + .XVBN^9,
3589      5124 6                  512, .COMPARE_BCB[BCB_BUFFER] + BBH$C_LENGTH + .XVBN^9)
3590      5125 6              THEN
3591      5126 6                  IF NOT FIND_BADBLOCK(.OUTPUT_BAD, .REC[BRH$L_ADDRESS] + .XVBN)
3592      5127 6                  THEN
3593      5128 6                      FILE_ERROR(
3594      5129 6                          BACKUP$_VERIFYERR,
3595      5130 6                          .OUTPUT_FAB,
3596      5131 6                          .REC[BRH$L_ADDRESS] + .XVBN);
3597      5132 5                      END;
3598      5133 5                  END
3599      5134 4              ELSE
3600      5135 5                  BEGIN
3601      5136 5                      ! Writing to an output file.
3602      5137 5
```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

E 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 117
(20)

```
: 3603      5138  5      !
: 3604      5139  5      BCB[BCB_RECORD] = .REC;
: 3605      5140  5      IF
: 3606      5141  6      BEGIN
: 3607      5142  6      LOCAL
: 3608      5143  6      LIMIT,
: 3609      5144  6      RLIMIT: REF BBLOCK;
: 3610      5145  6
: 3611      5146  6      LIMIT = .BCB[BCB_BUFFER] + .BCB[BCB_SIZE];
: 3612      5147  6      RLIMIT = .REC + .REC[BRH$W_RSIZE] + BRH$C_LENGTH;
: 3613      5148  6      IF .RLIMIT GEQA .LIMIT
: 3614      5149  6      THEN
: 3615      5150  6      TRUE
: 3616      5151  6      ELSE
: 3617      5152  6      .RLIMIT[BRH$W_RTYPE] EQL BRH$K_NULL AND
: 3618      5153  6      .RLIMIT + .RLIMIT[BRH$W_RSIZE] + BRH$C_LENGTH GEQA .LIMIT
: 3619      5154  6
: 3620      5155  5      END
: 3621      5156  6      THEN
: 3622      5157  6      BEGIN
: 3623      5158  6      ! This is the last record in the current buffer.
: 3624      5159  6      ! Execute the write asynchronously.
: 3625      5160  6      !
: 3626      5161  6      STATUS = CSQIO(
: 3627      5162  6      FUNC=.OUTPUT_FUNC,
: 3628      5163  6      CHAN=.OUTPUT_CHAN,
: 3629      5164  6      EFN=BCB_S_WRITE,
: 3630      5165  6      IOSB=BCB[BCB_IOSB],
: 3631      5166  6      P1=.REC + BRH$C_LENGTH,
: 3632      5167  6      P2=.THIS_BLOCKS^ 9,
: 3633      5168  6      P3=.THIS_MINVBN);
: 3634      5169  6      IF NOT .STATUS
: 3635      5170  6      THEN
: 3636      5171  6      FILE_ERROR(
: 3637      5172  6      BACKUP$_WRITEERR + ST$K_SEVERE,
: 3638      5173  6      .OUTPUT_FAB,
: 3639      5174  6      .STATUS);
: 3640      5175  6      BCB[BCB_SUCC_ACT] = 0;
: 3641      5176  6      BCB[BCB_FAIL_ACT] = RESTORE_ERROR;
: 3642      5177  6      BCB[BCB_STATE] = BCB_S_WRITE;
: 3643      5178  6      INSQUE(.BCB, .OUTPUT_WAIT[1]);
: 3644      5179  6      END
: 3645      5180  5      ELSE
: 3646      5181  6      BEGIN
: 3647      5182  6
: 3648      5183  6      ! This is not the last record in the buffer. Execute the
: 3649      5184  6      ! write synchronously.
: 3650      5185  6      !
: 3651      5186  6      STATUS = CSQIOW(
: 3652      5187  6      FUNC=.OUTPUT_FUNC,
: 3653      5188  6      CHAN=.OUTPUT_CHAN,
: 3654      5189  6      IOSB=IOSB,
: 3655      5190  6      P1=.REC + BRH$C_LENGTH,
: 3656      5191  6      P2=.THIS_BLOCKS^ 9,
: 3657      5192  6      P3=.THIS_MINVBN);
: 3658      5193  6      IF .STATUS THEN STATUS = .IOSB[0];
: 3659      5194  6      IF NOT .STATUS THEN RESTORE_ERROR(.BCB);
```

RESTORE
V04-G00

Restore a save set
RESTORE_ONE_RECORD - restore save set record

F 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 118
(20)

```
: 3660      5195  5      END;
: 3661      5196  4      END;
: 3662      5197  4
: 3663      5198  4
: 3664      5199  4      ! Update the output block number past this data record.
: 3665      5200  4      !
: 3666      5201  4      OUTPUT_BLOCK = .THIS_MAXVBN + 1;
: 3667      5202  3      END;
: 3668      5203  3
: 3669      5204  3
: 3670      5205  3      ! Indicate no longer the first data record of a verify pass.
: 3671      5206  3      !
: 3672      5207  3      OUTPUT_FLAGS[OUTPUT_V_FIRST] = FALSE;
: 3673      5208  2      END;
: 3674      5209  2
: 3675      5210  2
: 3676      5211  2      [BRH$K SUMMARY]:
: 3677      5212  2      BEGIN
: 3678      5213  3      LOCAL
: 3679      5214  3      SAVE_ATTBUF:      BBLOCK[FAR_LENGTH];
: 3680      5215  3
: 3681      5216  3
: 3682      5217  3      ! Save the contents of the attributes buffer. This is necessary
: 3683      5218  3      ! because the BACKUP summary record on a continuation tape is usually
: 3684      5219  3      ! interspersed with file data records.
: 3685      5220  3      !
: 3686      5221  3      CH$MOVE(FAR_LENGTH, OUTPUT_ATTBUF, SAVE_ATTBUF);
: 3687      5222  3
: 3688      5223  3
: 3689      5224  3      ! Get the attributes and save the volume set count, volume set name,
: 3690      5225  3      ! and creation date of the save set for later use.
: 3691      5226  3      !
: 3692      5227  3      GET_SUMMARY_ATTRIBUTES(.REC);
: 3693      5228  3      IF .QUAL[QUAL_OF11]
: 3694      5229  3      THEN
: 3695      5230  4      BEGIN
: 3696      5231  4      COM_O_SETCOUNT = .OUTPUT_ATTBUF[BSR_NVOLS];
: 3697      5232  4      COM_O_BSRDATE[0] = .(OUTPUT_ATTBUF[BSR_DATE]);
: 3698      5233  4      COM_O_BSRDATE[1] = .(OUTPUT_ATTBUF[BSR_DATE]+4);
: 3699      5234  4
: 3700      5235  4      IF
: 3701      5236  4      .QUAL[QUAL_VOLU] AND
: 3702      5237  4      .QUAL[QUAL_VOLU_VALUE] GTRU .COM_O_SETCOUNT
: 3703      5238  4      THEN
: 3704      5239  4      SIGNAL(
: 3705      5240  4      BACKUP$ NOSUCHRVN, 2,
: 3706      5241  4      .QUAL[QUAL_VOLU_VALUE], .COM_O_SETCOUNT);
: 3707      5242  4
: 3708      5243  4      CH$COPY(
: 3709      5244  4      .BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$W_LENGTH],
: 3710      5245  4      .BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$A_POINTER],
: 3711      5246  4      XC,
: 3712      5247  4      HM2$S_STRUCNAME,
: 3713      5248  4      COM_O_STRUCNAME);
: 3714      5249  3      END;
: 3715      5250  3
: 3716      5251  3
```


RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

G 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 119
(20)

```
: 3717 5252 3      ! Restore the attribute buffer.
: 3718 5253 3
: 3719 5254 3      CH$MOVE(FAR_LENGTH, SAVE_ATTBUF, OUTPUT_ATTBUF);
: 3720 5255 3      END;
: 3721 5256 2
: 3722 5257 2
: 3723 5258 2      [BRH$K_VOLUME]:
: 3724 5259 3      BEGIN
: 3725 5260 3
: 3726 5261 3      ! If this is an image restore or image copy, initialize the output
: 3727 5262 3      ! volume.
: 3728 5263 3
: 3729 5264 3      GET_VOLUME_ATTRIBUTES(.REC);
: 3730 5265 3      IF
: 3731 5266 3          .QUAL[QUAL_IMAG] AND
: 3732 5267 3          NOT .QUAL[QUAL_COMP] AND
: 3733 5268 4          (NOT .QUAL[QUAL_VOLU] OR
: 3734 5269 4          .QUAL[QUAL_VOLU_VALUE] EQL .OUTPUT_ATTBUF[VSR_RVN] OR
: 3735 5270 4          .QUAL[QUAL_VOLU_VALUE] EQL 1 AND .OUTPUT_ATTBUF[VSR_RVN] LEQU 1)
: 3736 5271 3      THEN
: 3737 5272 3          STA_INIVOL();
: 3738 5273 2      END;
: 3739 5274 2
: 3740 5275 2      [BRH$K_FID]:
: 3741 5276 2      BEGIN
: 3742 5277 3
: 3743 5278 3      ! Check the structure level word. If this is an image restore or
: 3744 5279 3      ! image copy, initialize the file headers that are described by
: 3745 5280 3      ! this record.
: 3746 5281 3
: 3747 5282 3
: 3748 5283 3      IF .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_STRUCLEV] NEQ BBH$K_LEVEL1
: 3749 5284 3      THEN
: 3750 5285 3          SIGNAL(BACKUP$_INVATTSTR)
: 3751 5286 3      ELSE
: 3752 5287 3          IF
: 3753 5288 3              .QUAL[QUAL_IMAG] AND
: 3754 5289 3              NOT .QUAL[QUAL_COMP] AND
: 3755 5290 4              (NOT .QUAL[QUAL_VOLU] OR
: 3756 5291 4              .QUAL[QUAL_VOLU_VALUE] EQL .OUTPUT_ATTBUF[VSR_RVN] OR
: 3757 5292 4              .QUAL[QUAL_VOLU_VALUE] EQL 1 AND .OUTPUT_ATTBUF[VSR_RVN] LEQU 1)
: 3758 5293 3          THEN
: 3759 5294 3              STA_INIT_HDRS(REC[BRH$C_LENGTH,0,0,0]);
: 3760 5295 2      END;
: 3761 5296 2
: 3762 5297 2
: 3763 5298 2      [BRH$K_PHYSVOL]:
: 3764 5299 3      BEGIN
: 3765 5300 3
: 3766 5301 3      ! Get the attributes. However, on a verify pass, do not reprocess
: 3767 5302 3      ! the record.
: 3768 5303 3
: 3769 5304 3      GET_PHYSVOL_ATTRIBUTES(.REC);
: 3770 5305 3      IF .COM_FLAGS[COM_VERIFYING] THEN RETURN;
: 3771 5306 3
: 3772 5307 3      ! If the NONSEQUENTIAL flag is set in the record header then
: 3773 5308 3      ! set our internal flag
```

```
3774 5309 3
3775 5310 3
3776 5311 3
3777 5312 3
3778 5313 3
3779 5314 3
3780 5315 3
3781 5316 3
3782 5317 3
3783 5318 3
3784 5319 3
3785 5320 3
3786 5321 3
3787 5322 3
3788 5323 3
3789 5324 3
3790 5325 3
3791 5326 3
3792 5327 3
3793 5328 3
3794 5329 3
3795 5330 3
3796 5331 4
3797 5332 4
3798 5333 4
3799 5334 4
3800 5335 5
3801 5336 5
3802 5337 5
3803 5338 5
3804 5339 6
3805 5340 6
3806 5341 6
3807 5342 6
3808 5343 6
3809 5344 7
3810 5345 7
3811 5346 7
3812 5347 7
3813 5348 7
3814 5349 7
3815 5350 7
3816 5351 7
3817 5352 6
3818 5353 5
3819 5354 6
3820 5355 6
3821 5356 6
3822 5357 5
3823 5358 4
3824 5359 4
3825 5360 3
3826 5361 2
3827 5362 2
3828 5363 2
3829 5364 2
3830 5365 3

!
IF .REC[BRH$V_NONSEQUENTIAL]
THEN
    OUTPUT_FLAGS[OUTPUT_NONSEQ_IN] = TRUE ;

! Make sure the device geometry matches.
IF
    .OUTPUT_ATTBUF[PVA_SECTORS] NEQ .OUTPUT_DEVGEOM[DGM_SECTORS] OR
    .OUTPUT_ATTBUF[PVA_TRACKS] NEQ .OUTPUT_DEVGEOM[DGM_TRACKS] OR
    .OUTPUT_ATTBUF[PVA_CYLINDERS] NEQ .OUTPUT_DEVGEOM[DGM_CYLINDERS] OR
    .OUTPUT_ATTBUF[PVA_MAXBLOCK] NEQ .OUTPUT_DEVGEOM[DGM_MAXBLOCK]
THEN
    FILE_ERROR(BACKUP$_PHYDEVGEO, .OUTPUT_FAB);

! Ensure that there are no bad blocks on the output device that were
! good blocks on the input device. Do this by matching each bad LBN
! in the output device data against the attribute record. If the LBN
! is not found, it is an error.
INCR I FROM 0 TO .OUTPUT_BAD[BAD_NUMDESC]-1 DO
    BEGIN
        LOCAL Q: REF BBLOCK;
        Q = OUTPUT_BAD[BAD_DESC];
        INCRU LBN FROM .Q[BAD_LBN] TO .Q[BAD_LBN]+.Q[BAD_COUNT]-1 DO
            BEGIN
                LOCAL P: REF VECTOR;
                P = .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSC$_POINTER];
                IF
                    BEGIN
                        WHILE .P LSSA
                            .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSC$_POINTER] +
                            .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSC$_LENGTH]
                        DO
                            BEGIN
                                IF
                                    .LBN GEQU .P[0] AND
                                    .LBN LEQU .P[0]+.P[1]-1
                                THEN
                                    EXITLOOP FALSE;
                                P = .P + 8;
                            END
                        END
                    THEN
                        BEGIN
                            FILE_ERROR(BACKUP$_BADBLOCK, .OUTPUT_FAB);
                            RETURN;
                        END;
                    END;
                Q = .Q + BAD_S_DESC;
            END;
        END;

[BRH$_FILE]:
BEGIN
```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

1 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 121
(20)

```

: 3831 5366 3 LOCAL
: 3832 5367 3 BUFFER: REF BBLOCK; ! Pointer to buffer
: 3833 5368 3
: 3834 5369 3
: 3835 5370 3 ! If a file is currently open, close it.
: 3836 5371 3 !
: 3837 5372 3 CLOSE_RESTORE();
: 3838 5373 3
: 3839 5374 3
: 3840 5375 3 ! Get the attributes from the record.
: 3841 5376 3 !
: 3842 5377 3 GET_FILE_ATTRIBUTES(.REC);
: 3843 5378 3 IF
: 3844 5379 3 .QUAL[QUAL_TRUN] AND
: 3845 5380 3 .OUTPUT_RECATTR[FAT$V_FILEORG] EQL FAT$C_SEQUENTIAL
: 3846 5381 3 THEN
: 3847 5382 4 BEGIN
: 3848 5383 4 LOCAL
: 3849 5384 4 EOF_SIZE;
: 3850 5385 4
: 3851 5386 4 EOF_SIZE = ROT(.OUTPUT_RECATTR[FAT$L_EFBLK], 16);
: 3852 5387 4 IF .OUTPUT_RECATTR[FAT$W_FFBYTE] EQL 0
: 3853 5388 4 THEN EOF_SIZE = .EOF_SIZE - 1;
: 3854 5389 4 OUTPUT_ATTBUF[FAR_FILESIZE] =
: 3855 5390 4 MINU(.EOF_SIZE, .OUTPUT_ATTBUF[FAR_FILESIZE]);
: 3856 5391 3 END;
: 3857 5392 3 BUFFER = .OUTPUT_BCB[BCB_BUFFER];
: 3858 5393 3 BUFFER[BBH$W_FID_NUM] = .OUTPUT_ATTBUF[FAR_FID_NUM];
: 3859 5394 3 BUFFER[BBH$W_FID_SEQ] = .OUTPUT_ATTBUF[FAR_FID_SEQ];
: 3860 5395 3 BUFFER[BBH$W_FID_RVN] = .OUTPUT_ATTBUF[FAR_FID_RVNW];
: 3861 5396 3
: 3862 5397 3
: 3863 5398 3 ! If this is a save verify pass, reassign the channel to the output
: 3864 5399 3 ! volume if required.
: 3865 5400 3 !
: 3866 5401 3 IF .COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]
: 3867 5402 3 THEN ASSIGN_VERIFY_CHANNEL(TRUE);
: 3868 5403 3
: 3869 5404 3
: 3870 5405 3 ! Determine if the new file is selected, and if it is, process it in
: 3871 5406 3 ! the appropriate manner.
: 3872 5407 3 !
: 3873 5408 3 IF INITIALIZE_AND_SELECT()
: 3874 5409 3 THEN
: 3875 5410 4 BEGIN
: 3876 5411 4 IF .REC[BRH$V_DIRECTORY]
: 3877 5412 4 THEN
: 3878 5413 4 IF .QUAL[QUAL_COMP]
: 3879 5414 4 THEN
: 3880 5415 4 IF
: 3881 5416 4 (.QUAL[QUAL_ISAV] AND .QUAL[QUAL_IMAG]) OR
: 3882 5417 4 (.COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV])
: 3883 5418 4 THEN
: 3884 5419 4 CREATE_OR_OPEN_FILE() ! compare directory contents
: 3885 5420 4 ELSE
: 3886 5421 4 0 ! skip directory
: 3887 5422 4 ELSE
```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

J 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 122
(20)

```

3888      5423  4      IF .QUAL[QUAL_IMAG]
3889      5424  4      THEN
3890      5425  4          CREATE_OR_OPEN_FILE()      ! image restore directory
3891      5426  4      ELSE
3892      5427  4          CREATE_DIRECTORY()          ! non-image restore directory
3893      5428  4      ELSE
3894      5429  4          IF
3895      5430  5              BEGIN
3896      5431  5                  IF NOT .QUAL[QUAL_INCR]
3897      5432  5                      THEN TRUE
3898      5433  5                      ELSE SELECT_INCREMENTAL()
3899      5434  5                      END
3900      5435  4              THEN
3901      5436  4                  CREATE_OR_OPEN_FILE();      ! compare/restore file
3902      5437  4
3903      5438  4      ! Initialize the ACL restoration FIB.
3904      5439  4      !
3905      5440  4      CH$FILL (0, FIB$C_LENGTH, ACL_FIB); ! Reset the FIB
3906      5441  4      ACL_FIB[FIB$V_NORECORD] = TRUE;
3907      5442  4      IF .QUAL[QUAL_IMAG]
3908      5443  4      THEN
3909      5444  4          BEGIN
3910      5445  5              ACL_FIB[FIB$W_FID_NUM] = .OUTPUT_ATTBUF[FAR_FID_NUM];
3911      5446  5              ACL_FIB[FIB$W_FID_SEQ] = .OUTPUT_ATTBUF[FAR_FID_SEQ];
3912      5447  5              ACL_FIB[FIB$W_FID_RVN] = .OUTPUT_ATTBUF[FAR_FID_RVN];
3913      5448  5              END
3914      5449  5          ELSE
3915      5450  5              BEGIN
3916      5451  5                  ACL_FIB[FIB$W_FID_NUM] = .OUTPUT_NAM[NAM$W_FID_NUM];
3917      5452  5                  ACL_FIB[FIB$W_FID_SEQ] = .OUTPUT_NAM[NAM$W_FID_SEQ];
3918      5453  5                  ACL_FIB[FIB$W_FID_RVN] = .OUTPUT_NAM[NAM$W_FID_RVN];
3919      5454  5                  END;
3920      5455  4          OUTPUT_FLAGS[ACL_ERROR] = FALSE;      ! No error processing ACL yet
3921      5456  4          OUTPUT_FLAGS[ACL_FIRST_TIME] = TRUE;      ! First time through
3922      5457  4          CH$FILE (0, 8, ACL_FIB_DESCR);
3923      5458  4          ACL_FIB_DESCR[DSC$W_LENGTH] = FIB$C_LENGTH;
3924      5459  4          ACL_FIB_DESCR[DSC$A_POINTER] = ACL_FIB;
3925      5460  4
3926      5461  4      ! If there is an ACL to restore, restore it.
3927      5462  4      !
3928      5463  4      IF (.OUTPUT_ACLSEGMENT[DSC$W_LENGTH] NEQ 0)
3929      5464  4      AND (NOT .COM_FLAGS[COM_VERIFYING] )
3930      5465  4      AND (.OUTPUT_FLAGS[OUTPUT_OPEN] OR .OUTPUT_FLAGS[OUTPUT_INCR_DIR]
3931      5466  4          OR .OUTPUT_FLAGS[OUTPUT_DIR_OPEN] )
3932      5467  4      AND NOT .QUAL[QUAL_INTE]
3933      5468  4      THEN
3934      5469  4          BEGIN
3935      5470  5              LOCAL
3936      5471  5                  ACL_POINTER : REF BBLOCK,      ! Pointer to ACE within ACL
3937      5472  5                  ATR_DESC      : BBLOCK [12],      ! ACP Attribute list
3938      5473  5                  IOSB        : VECTOR [4,WORD],      ! I/O status block
3939      5474  5                  STATUS      : BBLOCK [4],      ! Poutine exit status
3940      5475  5                  DUMMY_ACL   : BBLOCK [8];      ! Dummy ACL for delete operation
3941      5476  5
3942      5477  5      ! If this is the first ACL record, delete any existing ACL on
3943      5478  5
3944      5479  5
```

```

3945      5480 5      ! the selected file to avoid confusion.
3946      5481 5
3947      5482 5
3948      5483 5
3949      5484 6
3950      5485 6      OUTPUT_FLAGS[ACL_FIRST_TIME] = FALSE;
3951      5486 6      ATR_DESC[0,0,16,0] = 8;
3952      5487 6      ATR_DESC[2,0,16,0] = ATR$C_DELETEACL;
3953      5488 6      ATR_DESC[4,0,32,0] = DUMMY_ACL;
3954      5489 6      ATR_DESC[8,0,32,0] = 0;
3955      5490 6
3956      P      5491 6      STATUS = C$QIOW (CHAN = .OUTPUT_CHAN,
3957      P      5492 6      FUNC = IOS$MODIFY,
3958      P      5493 6      IOSB = IOSB,
3959      P      5494 6      P1 = ACL_FIB_DESCR,
3960      5495 6      P5 = ATR_DESC);
3961      5496 6      IF .STATUS THEN STATUS = .IOSB[0];
3962      5497 6      IF NOT .STATUS
3963      5498 6      THEN
3964      5499 7      BEGIN
3965      5500 7      FILE_ERROR (
3966      5501 7      BACKUPS$WRITEERR + STS$K_WARNING,
3967      5502 7      .OUTPUT_FAB,
3968      5503 7      .STATUS);
3969      5504 7      OUTPUT_FLAGS[ACL_ERROR] = TRUE; ! Note delete failed
3970      5505 7      RETURN;
3971      5506 6      END;
3972      5507 5      END;
3973      5508 5
3974      5509 5      ! If any errors occurred deleting or writing the file's ACL,
3975      5510 5      ! ignore any further ACL records for the file.
3976      5511 5
3977      5512 5      IF .OUTPUT_FLAGS[ACL_ERROR] THEN RETURN;
3978      5513 5
3979      5514 5      ! Add the ACE's to the selected file's ACL
3980      5515 5
3981      5516 5      ATR_DESC[0,0,16,0] = .OUTPUT_ACLSEGMENT[DSC$W_LENGTH];
3982      5517 5      ATR_DESC[2,0,16,0] = ATR$C_ADDACLNT;
3983      5518 5      ATR_DESC[4,0,32,0] = .OUTPUT_ACLSEGMENT[DSC$A_POINTER];
3984      5519 5      ATR_DESC[8,0,32,0] = 0;
3985      P      5520 5      STATUS = C$QIOW (CHAN = .OUTPUT_CHAN,
3986      P      5521 5      FUNC = IOS$MODIFY,
3987      P      5522 5      IOSB = IOSB,
3988      P      5523 5      P1 = ACL_FIB_DESCR,
3989      5524 5      P5 = ATR_DESC);
3990      5525 5      IF .STATUS THEN STATUS = .IOSB[0];
3991      5526 5      IF NOT .STATUS
3992      5527 5      THEN
3993      5528 6      BEGIN
3994      5529 6      FILE_ERROR (
3995      5530 6      BACKUPS$WRITEERR + STS$K_WARNING,
3996      5531 6      .OUTPUT_FAB,
3997      5532 6      .STATUS);
3998      5533 6      OUTPUT_FLAGS[ACL_ERROR] = TRUE; ! Note add failed
3999      5534 6      RETURN;
4000      5535 5      END;
4001      5536 5      OUTPUT_ACLSEGMENT[DSC$W_LENGTH] = 0; ! Note segment added to ACL

```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

L 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 124
(20)

```

4002 5537 4
4003 5538 4
4004 5539 4
4005 5540 4
4006 5541 4
4007 5542 4
4008 5543 4
4009 5544 2
4010 5545 2
4011 5546 2
4012 5547 2
4013 5548 4
4014 5549 4
4015 5550 4
4016 5551 4
4017 5552 4
4018 5553 4
4019 5554 4
4020 5555 4
4021 5556 4
4022 5557 4
4023 5558 4
4024 5559 4
4025 5560 4
4026 5561 4
4027 5562 3
4028 5563 3
4029 5564 4
4030 5565 4
4031 5566 4
4032 5567 4
4033 5568 4
4034 5569 4
4035 5570 4
4036 5571 4
4037 5572 4
4038 5573 4
4039 5574 4
4040 5575 4
4041 5576 4
4042 5577 5
4043 5578 5
4044 5579 5
4045 5580 5
4046 5581 5
4047 5582 5
4048 5583 5
4049 5584 5
4050 5585 5
4051 5586 5
4052 5587 5
4053 5588 5
4054 5589 5
4055 5590 5
4056 5591 5
4057 5592 6
4058 5593 6

      END;
    END;

    ! Indicate no longer the first data record of a verify pass.
    !
    OUTPUT_FLAGS[OUTPUT_V_FIRST] = FALSE;
  END;

[BRH$K_FILE_EXT]:
  BEGIN

    ! Get any additional file attributes from the record.
    !
    GET_FILE_EXT_ATTRIBUTES (.REC);

    ! If there is an ACL to restore, restore it.
    !
    IF (.OUTPUT_ACLSEGMENT[DSC$W_LENGTH] NEQ 0)
    AND (NOT .COM_FLAGS[COM_VERIFYING])
    AND (.OUTPUT_FLAGS[OUTPUT_OPEN] OR .OUTPUT_FLAGS[OUTPUT_INCR_DIR]
    OR .OUTPUT_FLAGS[OUTPUT_DIR_OPEN])
    AND NOT .QUAL[QUAL_INTE]
    THEN
      BEGIN
        LOCAL
          ACL_POINTER      : REF BBLOCK,      ! Pointer to ACE within ACL
          ATR_DESC         : BBLOCK [12],     ! ACP Attribute list
          IOSB              : VECTOR [4,WORD], ! I/O status block
          STATUS            : ROUTINE_EXIT_STATUS, ! Routine exit status
          DUMMY_ACL         : BBLOCK [8];      ! Dummy ACL for delete operation

        ! If this is the first ACL record, delete any existing ACL on
        ! the selected file to avoid confusion.
        !
        IF .OUTPUT_FLAGS[ACL_FIRST_TIME]
        THEN
          BEGIN
            OUTPUT_FLAGS[ACL_FIRST_TIME] = FALSE;
            ATR_DESC[0,0,16,0] = 8;
            ATR_DESC[2,0,16,0] = ATR$C_DELETEACL;
            ATR_DESC[4,0,32,0] = DUMMY_ACL;
            ATR_DESC[8,0,32,0] = 0;

            STATUS = C$QIOW (CHAN = .OUTPUT_CHAN,
                           FUNC = IOSB_MODIFY,
                           IOSB = IOSB,
                           P1 = ACL_FIB_DESCR,
                           P5 = ATR_DESC);

            IF .STATUS THEN STATUS = .IOSB[0];
            IF NOT .STATUS
            THEN
              BEGIN
                FILE_ERROR (

```

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

M 13
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 125
(20)

```

: 4059      5594 6      BACKUPS_WRITEERR + STS$K_WARNING,
: 4060      5595 6      .OUTPUT_FAB,
: 4061      5596 6      .STATUS;
: 4062      5597 6      OUTPUT_FLAGS[ACL_ERROR] = TRUE;      ! Note delete failed
: 4063      5598 6      RETURN;
: 4064      5599 5      END;
: 4065      5600 4      END;
: 4066      5601 4
: 4067      5602 4      ! If any errors occurred deleting or writing the file's ACL,
: 4068      5603 4      ! ignore any further ACL records for the file.
: 4069      5604 4
: 4070      5605 4      IF .OUTPUT_FLAGS[ACL_ERROR] THEN RETURN;
: 4071      5606 4
: 4072      5607 4      ! Add the ACE's to the selected file's ACL
: 4073      5608 4
: 4074      5609 4      ATR_DESC[0,0,16,0] = .OUTPUT_ACLSEGMENT[DSC$W_LENGTH];
: 4075      5610 4      ATR_DESC[2,0,16,0] = ATR$C_ADDACLENT;
: 4076      5611 4      ATR_DESC[4,0,32,0] = .OUTPUT_ACLSEGMENT[DSC$A_POINTER];
: 4077      5612 4      ATR_DESC[8,0,32,0] = 0;
: 4078      5613 4      STATUS = C$QIOW (CHAN = .OUTPUT_CHAN,
: 4079      5614 4      FUNC = IO$MODIFY,
: 4080      5615 4      IOSB = IOSB,
: 4081      5616 4      P1 = ACL_FIB_DESCR,
: 4082      5617 4      P5 = ATR_DESC);
: 4083      5618 4      IF .STATUS THEN STATUS = .IOSB[0];
: 4084      5619 4      IF NOT .STATUS
: 4085      5620 4      THEN
: 4086      5621 5      BEGIN
: 4087      5622 5      FILE_ERROR (
: 4088      5623 5      BACKUPS_WRITEERR + STS$K_WARNING,
: 4089      5624 5      .OUTPUT_FAB,
: 4090      5625 5      .STATUS;
: 4091      5626 5      OUTPUT_FLAGS[ACL_ERROR] = TRUE; ! Note add failed
: 4092      5627 5      RETURN;
: 4093      5628 4      END;
: 4094      5629 4      OUTPUT_ACLSEGMENT[DSC$W_LENGTH] = 0;      ! Note segment added to ACL
: 4095      5630 3      END;
: 4096      5631 2      END;
: 4097      5632 2
: 4098      5633 2
: 4099      5634 2      TES;
: 4100      5635 1      END;
```

				OFFC 00000	.ENTRY	RESTORE ONE RECORD, Save R2,R3,R4,R5,R6,R7,-;	4854
						R8,R9,RT0,RT1	
		5E	FF6C	CE 9E 00002	MOVAB	-148(SP), SP	
		57	04	AC 7D 00007	MOVQ	REC, R7	4891
		00000000		58 D0 0000B	MOVL	R8, OUTPUT_BCB	4888
		01	02	A7 AF 00012	CASEW	2(R7), #1, #7	4891
0011	068A	0508	0487	00017 1\$:	.WORD	42\$-1\$,-	
091E	0552	0011	05B0	0001F		45\$-1\$,-	
						69\$-1\$,-	
						2\$-1\$,-	

```
56$-1$,-
2$-1$,-
47$-1$,-
95$-1$,-
50  A9 00000000' 59 0C A8 D0 00027 2$: RET
0E 00000000' EF 06 29 00028 MOVL 12(R8), R9 4908
06 00000000' EF 03 12 0002C CMPC3 #6, OUTPUT_ATTBUF+80, 80(R9)
01 22 A9 B1 00035 BNEQ 3$
00000000' EF DD 00037 BRW 9$
00000000G 00 01 E1 0003A 3$: BBC #1, OUTPUT_FLAGS, 4$ 4919
09 EB08 CF 03 E1 00042 BBC #3, COM_FLAGS, 4$ 4920
01 00000000' EF 01 13 1A 0004A CMPW 34(R9), #1 4921
00000000' EF DD 00050 4$: PUSHL OUTPUT_FAB 4925
00000000G 00 8F DD 00056 PUSHL #BACKUP$ FIDERROR 4923
00000000' CF 02 FB 0005C CALLS #2, FILE_ERROR
09 04 A7 00 01 FB 00063 5$: CALLS #0, CLOSE_RESTORE 4930
01 00000000' EF 03 E0 00068 BBC #1, 4(R7), 6$ 4936
01 00000000' EF 04 00075 RET
01 00000000' EF E9 00076 6$: BLBC QUAL+14, 7$ 4937
01 00000000' EF 04 E1 0007D RET
04 0007E 7$: BBC #4, QUAL+10, 8$ 4938
04 00086 RET
0090 8F 00 56 59 D0 00087 8$: MOVL R9, BUFFER 4946
6E 00 2C 0008A MOVCS #0, (SP), #0, #144, OUTPUT_ATTBUF 4947
00000000' EF 00091
00000000' EF 50 A6 D0 00096 MOVL 80(BUFFER), OUTPUT_ATTBUF+80 4948
00000000' EF 54 A6 B0 0009E MOVW 84(BUFFER), OUTPUT_ATTBUF+84 4950
00000000' EF 5C A6 9B 000A6 MOVZBW 92(BUFFER), OUTPUT_FILENAME 4951
00000000' EF 5D A6 9E 000AE MOVAB 93(R6), OUTPUT_FILENAME+4 4952
00000000' EF 00DC C6 D0 000B6 MOVL 220(BUFFER), OUTPUT_RECATTR 4953
50 00E4 C6 01 C1 000BF ADDL3 #1, 228(BUFFER), R0 4956
00000000' EF 50 10 9C 000C5 ROTL #16, R0, OUTPUT_RECATTR+8
00000000' EF 00E0 C6 D0 000CD MOVL 224(BUFFER), OUTPUT_RECATTR+14 4957
00000000' EF 00E4 C6 D0 000D6 MOVL 228(BUFFER), OUTPUT_ATTBUF+92 4960
F2BB CF 00 FB 000DF CALLS #0, INITIALIZE_AND_SELECT 4966
13 50 E9 000E4 BLBC R0, 9$
F8CA CF 00 FB 000E7 CALLS #0, CREATE_OR_OPEN_FILE 4969
07 00000000' EF E9 000EC BLBC OUTPUT_FLAGS, 9$ 4970
00000000' EF 10 88 000F3 BISB2 #16, OUTPUT_FLAGS 4971
0B 00000000' EF 03 E1 000FA 9$: BBC #3, OUTPUT_FLAGS, 10$ 4978
10 A7 9F 00102 PUSHAB 16(R7) 4980
7E 67 3C 00105 MOVZWL (R7), -(SP)
07 F6E6 CF 02 FB 00108 CALLS #2, INCREMENTAL_DIRECTORY
04 A7 02 E1 0010D BBC #2, 4(R7), 11$ 4986
00000000' EF 01 88 00112 BISB2 #1, OUTPUT_FLAGS+1 4988
03 00000000' EF E8 00119 11$: BLBS OUTPUT_FLAGS, 12$ 4993
080A 31 00120 BRW 94$
5A 08 A7 D0 00123 12$: MOVL 8(R7), THIS_MINVBN 5006
5B 67 3C 00127 MOVZWL (R7), R11 5007
55 5B F7 8F 78 0012A ASHL #9, R11, THIS_BLOCKS
00000000' EF 6E FF A54A 9E 0012F MOVAB -1(THIS_BLOCKS)[THIS_MINVBN], THIS_MAXVBN 5008
5A D1 00134 CMPL THIS_MINVBN, OUTPUT_BLOCK 5016
24 1E 0013B BGEQU 13$
7E 00000000' 1D 00000000' EF E8 0013D BLBS OUTPUT_FLAGS+1, 13$ 5017
EF 01 C3 00144 SUBL3 #1, OUTPUT_BLOCK, -(SP) 5023
```


			5A DD 0014C	PUSHL	THIS MINVBN	
		00000000G	EF DD 0014E	PUSHL	OUTPUT FAB	5022
		00000000G	8F DD 00154	PUSHL	#BACKUP\$ VBNERROR	5020
00000000G	00		04 FB 0015A	CALLS	#4, FILE_ERROR	
	42	00000000'	EF E8 00161	BLBS	QUAL+14, 15\$	5035
00000000'	EF		5A D1 00168	CMPL	THIS_MINVBN, OUTPUT_BLOCK	5036
			39 1B 0016F	BLEQU	15\$	
0E 00000000'	EF		01 E1 00171	BBC	#1, OUTPUT_FLAGS, 14\$	5037
06 00000000'	EF		03 E1 00179	BBC	#3, COM_FLAGS, 14\$	5038
	01	22	A9 B1 00181	CMPL	34(R9), -#1	5039
			23 1A 00185	BGTRU	15\$	
	1C	00000000'	EF E8 00187	BLBS	OUTPUT_FLAGS+1, 15\$	5040
		FF	AA 9F 0018E	PUSHAB	-1(THIS_MINVBN)	5046
		00000000'	EF DD 00191	PUSHL	OUTPUT_BLOCK	
		00000000'	EF DD 00197	PUSHL	OUTPUT_FAB	5045
		00000000G	8F DD 0019D	PUSHL	#BACKUP\$ VBNMISSING	5043
00000000G	00		04 FB 001A3	CALLS	#4, FILE_ERROR	
00000000'	EF		6E D1 001AA	CMPL	THIS_MAXVBN, OUTPUT_MAXBLOCK	5054
			2E 1B 001B1	BLEQU	17\$	
			6E DD 001B3	PUSHL	THIS_MAXVBN	5060
7E 00000000'	EF		01 C1 001B5	ADDL3	#1, OUTPUT_MAXBLOCK, -(SP)	
		00000000'	EF DD 001BD	PUSHL	OUTPUT_FAB	5059
		00000000G	8F DD 001C3	PUSHL	#BACKUP\$ VBNPASTEOF	5057
00000000G	00		04 FB 001C9	CALLS	#4, FILE_ERROR	
50 00000000'	EF		5A C3 001D0	SUBL3	THIS_MINVBN, OUTPUT_MAXBLOCK, R0	5061
			50 D6 001D8	INCL	R0	
			02 18 001DA	BGEQ	16\$	
			50 D4 001DC	CLRL	R0	
	55		50 D0 001DE	MOVL	R0, THIS_BLOCKS	
	1D	04	A7 E9 001E1	BLBC	4(R7), 18\$	5068
15 00000000'	EF		04 E0 001E5	BBS	#4, QUAL+15, 18\$	
			5A DD 001ED	PUSHL	THIS_MINVBN	5073
		00000000'	EF DD 001EF	PUSHL	OUTPUT_FAB	5072
		00000000G	8F DD 001F5	PUSHL	#BACKUP\$ BADDATA	5070
00000000G	00		03 FB 001FB	CALLS	#3, FILE_ERROR	
	56	00000000'	EF D0 00202	MOVL	OUTPUT_CHAN, R6	5091
59	55		09 78 00209	ASHL	#9, THIS_BLOCKS, R9	
		00000000'	EF 95 0020D	TSTB	QUAL+8	5076
			03 19 00213	BLSS	19\$	
			0183 31 00215	BRW	32\$	
		00000000'	EF D0 00218	MOVL	COMPARE_BCB, R0	5091
0001FFFF	8F		56 D1 0021F	CMPL	R6, #13T071	
			29 1F 00226	BLSSU	20\$	
			7E 7C 00228	CLRQ	-(SP)	
			7E D4 0022A	CLRL	-(SP)	
	7E		59 7D 0022C	MOVQ	R9, -(SP)	
7E	0C	A0 00000100	8F C1 0022F	ADDL3	#256, 12(R0), -(SP)	
			7E 7C 00238	CLRQ	-(SP)	
		F8	AD 9F 0023A	PUSHAB	IOSB	
		7E 00000000'	EF 9A 0023D	MOVZBL	INPUT_FUNC, -(SP)	
			56 DD 00244	PUSHL	R6	
			7E D4 00246	CLRL	-(SP)	
00000000G	00		0C FB 00248	CALLS	#12, STA_Q10W	
			27 11 0024F	BRB	21\$	
			7E 7C 00251	CLRQ	-(SP)	20\$:
			7E D4 00253	CLRL	-(SP)	
	7E		59 7D 00255	MOVQ	R9, -(SP)	

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

C 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 128
(20)

7E	0C	A0	00000100	8F	C1	00258	ADDL3	#256, 12(R0), -(SP)	
				7E	7C	00261	CLRQ	-(SP)	
			F8	AD	9F	00263	PUSHAB	IOSB	
		7E	00000000'	EF	9A	00266	MOVZBL	INPUT_FUNC, -(SP)	
				56	DD	0026D	PUSHL	R6	
				7E	D4	0026F	CLRL	-(SP)	
	00000000G	9F		0C	FB	00271	CALLS	#12, a#SYSSQIOW	
		54		50	DD	00278	MOVL	R0, STATUS	
		0A		54	E9	0027B	BLBC	STATUS, 22\$	5092
		54	F8	AD	3C	0027E	MOVZWL	IOSB, STATUS	
		03		54	E9	00282	BLBC	STATUS, 22\$	5093
				00C0	31	00285	BRW	29\$	
		52		01	CE	00288	MNEGL	#1, XVBN	5096
				00B1	31	0028B	BRW	27\$	
		50	00000000'	EF	DD	0028E	MOVL	COMPARE BCB, R0	5104
5B		52		09	78	00295	ASHL	#9, XVBN, R11	
		51	00000000'	EF	DD	00299	MOVL	OUTPUT_CHAN, R1	
	0001FFFF	8F		51	D1	002A0	CMPL	R1, #131071	
				32	1F	002A7	BLSSU	24\$	
				7E	7C	002A9	CLRQ	-(SP)	
				7E	D4	002AB	CLRL	-(SP)	
53		52	08	A7	C1	002AD	ADDL3	8(R7), XVBN, R3	
				53	DD	002B2	PUSHL	R3	
		7E	0200	8F	3C	002B4	MOVZWL	#512, -(SP)	
50		5B	0C	A0	C1	002B9	ADDL3	12(R0), R11, R0	
			0100	C0	9F	002BE	PUSHAB	256(R0)	
				7E	7C	002C2	CLRQ	-(SP)	
			F8	AD	9F	002C4	PUSHAB	IOSB	
		7E	00000000'	EF	9A	002C7	MOVZBL	INPUT_FUNC, -(SP)	
				51	DD	002CE	PUSHL	R1	
				7E	D4	002D0	CLRL	-(SP)	
	00000000G	00		0C	FB	002D2	CALLS	#12, STA_QIOW	
				30	11	002D9	BRB	25\$	
				7E	7C	002DB	CLRQ	-(SP)	
				7E	D4	002DD	CLRL	-(SP)	
53		52	08	A7	C1	002DF	ADDL3	8(R7), XVBN, R3	
				53	DD	002E4	PUSHL	R3	
		7E	0200	8F	3C	002E6	MOVZWL	#512, -(SP)	
50		5B	0C	A0	C1	002EB	ADDL3	12(R0), R11, R0	
			0100	C0	9F	002F0	PUSHAB	256(R0)	
				7E	7C	002F4	CLRQ	-(SP)	
			F8	AD	9F	002F6	PUSHAB	IOSB	
		7E	00000000'	EF	9A	002F9	MOVZBL	INPUT_FUNC, -(SP)	
				51	DD	00300	PUSHL	R1	
				7E	D4	00302	CLRL	-(SP)	
	00000000G	9F		0C	FB	00304	CALLS	#12, a#SYSSQIOW	
		54		50	DD	0030B	MOVL	R0, STATUS	
		07		54	E9	0030E	BLBC	STATUS, 26\$	5105
		54	F8	AD	3C	00311	MOVZWL	IOSB, STATUS	
		27		54	E8	00315	BLBS	STATUS, 27\$	5106
				53	DD	00318	PUSHL	R3	
			00000000'	EF	DD	0031A	PUSHL	OUTPUT_BAD	
	00000000G	00		02	FB	00320	CALLS	#2, FIND_BADBLOCK	
		15		50	E8	00327	BLBS	R0, 27\$	
				18	BB	0032A	PUSHR	#*M<R3,R4>	5111
			00000000'	EF	DD	0032C	PUSHL	OUTPUT_FAB	5110
			00000000G	8F	DD	00332	PUSHL	#BACKUP\$_READBLOCK	5108

02	00000000G	00	04	FB	00338	CALLS	#4, FILE_ERROR	5096
		52	55	F2	0033F	AOBLSS	THIS_BLOCKS, XVBN, 28\$	
			03	11	00343	BRB	29\$	
			FF46	31	00345	BRW	23\$	
		56	01	CE	00348	MNEGL	#1, XVBN	5120
			47	11	00348	BRB	31\$	
51		56	09	78	0034D	ASHL	#9, XVBN, R1	5123
		50	EF	DD	00351	MOVL	COMPARE_BCB, R0	5124
50		51	0C	A0	C1	ADDL3	12(R0), R1, R0	
0100	C0	10	A147	0200	8F	CMPC3	#512, 16(R1)[R7], 256(R0)	5122
			2B	13	00367	BEQL	31\$	
			08	B746	9F	PUSHAB	@8(R7)[XVBN]	5126
			00000000	EF	DD	PUSHL	OUTPUT_BAD	
		00	02	FB	00373	CALLS	#2, FIND_BADBLOCK	
		17	50	E8	0037A	BLBS	R0, 31\$	
			08	B746	9F	PUSHAB	@8(R7)[XVBN]	5131
			00000000	EF	DD	PUSHL	OUTPUT_FAB	5130
			00000000G	8F	DD	PUSHL	#BACKUP\$_VERIFYERR	5128
B5	00000000G	00	03	FB	0038D	CALLS	#3, FILE_ERROR	
		56	55	F2	00394	AOBLSS	THIS_BLOCKS, XVBN, 30\$	5120
			00F8	31	00398	BRW	41\$	5076
		10	A8	57	DD	MOVL	R7, 16(R8)	5139
			53	A7	9E	MOVAB	16(R7), R3	5168
			51	00000000	EF	MOVZBL	OUTPUT_FUNC, R1	
			52	08	A8	MOVZWL	8(R8), LIMIT	5146
			52	0C	A8	ADDL2	12(R8), LIMIT	
			50	10	AB47	MOVAB	16(R11)[R7], RLIMIT	5147
			52	50	D1	CMPL	RLIMIT, LIMIT	5148
				12	1E	BGEQU	33\$	
			02	A0	B5	TSTW	2(RLIMIT)	5152
				7F	12	BNEQ	37\$	
		55	60	3C	003C1	MOVZWL	(RLIMIT), R5	5153
		50	10	A540	9E	MOVAB	16(R5)[RLIMIT], R0	
		52	50	D1	003C9	CMPL	R0, LIMIT	
			72	1F	003CC	BLSSU	37\$	
		50	18	A8	9E	MOVAB	24(R8), R0	5168
	0001FFFF	8F	56	D1	003D2	CMPL	R6, #131071	
			1B	1F	003D9	BLSSU	34\$	
			7E	7C	003DB	CLRQ	-(SP)	
			7E	D4	003DD	CLRL	-(SP)	
			0608	8F	BB	PUSHR	#*M<R3,R9,R10>	
				7E	7C	CLRQ	-(SP)	
				50	DD	PUSHL	R0	
				51	DD	PUSHL	R1	
				56	DD	PUSHL	R6	
			02	DD	003EB	PUSHL	#2	
	00000000G	00	0C	FB	003ED	CALLS	#12, STA_QIO	
			19	11	003F4	BRB	35\$	
			7E	7C	003F6	CLRQ	-(SP)	
			7E	D4	003F8	CLRL	-(SP)	
			0608	8F	BB	PUSHR	#*M<R3,R9,R10>	
				7E	7C	CLRQ	-(SP)	
				50	DD	PUSHL	R0	
				51	DD	PUSHL	R1	
				56	DD	PUSHL	R6	
			02	DD	00406	PUSHL	#2	
	00000000G	9F	0C	FB	00408	CALLS	#12, @#SYSSQIO	

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

E 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 130
(20)

54	50	D0	0040F	35\$:	MOVL	R0, STATUS	:	5169		
15	54	E8	00412		BLBS	STATUS, 36\$	:	5174		
	54	DD	00415		PUSHL	STATUS	:	5173		
	EF	DD	00417		PUSHL	OUTPUT_FAB	:	5172		
00000000G	00	8F	DD	0041D	PUSHL	#BACKUP\$ WRITEERR+4	:			
		03	FB	00423	CALLS	#3, FILE_ERROR	:			
	20	A8	D4	0042A	36\$:	CLRL	32(R8)	5175		
24	A8	CF	9E	0042D	MOVAB	RESTORE_ERROR, 36(R8)	:	5176		
0A	A8	02	90	00433	MOVB	#2, 10(R8)	:	5177		
00000000'	FF	68	0E	00437	INSQUE	(R8), @OUTPUT_WAIT+4	:	5178		
		53	11	0043E	BRB	41\$	:	5140		
0001FFFF	8F	56	D1	00440	37\$:	CMPL	R6, #131071	5192		
		1C	1F	00447	BLSSU	38\$	:			
		7E	7C	00449	CLRQ	-(SP)	:			
		7E	D4	0044B	CLRL	-(SP)	:			
	0608	8F	BB	0044D	PUSHR	#M<R3,R9,R10>	:			
	F8	7E	7C	00451	CLRQ	-(SP)	:			
		AD	9F	00453	PUSHAB	IOSB	:			
		51	DD	00456	PUSHL	R1	:			
		56	DD	00458	PUSHL	R6	:			
		7E	D4	0045A	CLRL	-(SP)	:			
00000000G	00	0C	FB	0045C	CALLS	#12, STA_QIOW	:			
		1A	11	00463	BRB	39\$	:			
		7E	7C	00465	38\$:	CLRQ	-(SP)			
		7E	D4	00467	CLRL	-(SP)	:			
	0608	8F	BB	00469	PUSHR	#M<R3,R9,R10>	:			
	F8	7E	7C	0046D	CLRQ	-(SP)	:			
		AD	9F	0046F	PUSHAB	IOSB	:			
		51	DD	00472	PUSHL	R1	:			
		56	DD	00474	PUSHL	R6	:			
		7E	D4	00476	CLRL	-(SP)	:			
00000000G	9F	0C	FB	00478	CALLS	#12, @#SYSSQIOW	:			
54		50	D0	0047F	39\$:	MOVL	R0, STATUS	5193		
07		54	E9	00482	BLBC	STATUS, 40\$	:			
54	F8	AD	3C	00485	MCVZWL	IOSB, STATUS	:	5194		
07		54	E8	00489	BLBS	STATUS, 41\$	:			
		58	DD	0048C	40\$:	PUSHL	R8			
	FAB1	01	FB	0048E	CALLS	#1, RESTORE_ERROR	:			
00000000'	EF	01	C1	00493	41\$:	ADDL3	#1, TH'S_MAXVBN, OUTPUT_BLOCK	5201		
		048F	31	0049B	BRW	94\$	:	5207		
04	AE	8F	28	0049E	42\$:	MOVCS	#144, OUTPUT_ATTBUF, SAVE_ATTBUF	5221		
		57	DD	004A9	PUSHL	R7	:	5227		
	00000000G	00	01	FB	004AB	CALLS	#1, GET SUMMARY ATTRIBUTES			
59	00000000'	EF	06	E1	004B2	BBC	#6, QUAC+15, 44\$	5228		
00000000'	EF	00000000'	EF	90	004BA	MOVB	OUTPUT_ATTBUF+114, COM_O_SETCOUNT	5231		
00000000'	EF	00000000'	EF	7D	004C5	MOVQ	OUTPUT_ATTBUF+36, COM_O_BSRDATE	5232		
00000000'	2A	00000000'	EF	E9	004D0	BLBC	QUAL+14, 43\$	5236		
	00000000'	EF	00000000'	EF	91	004D7	CMPB	QUAL+79, COM_O_SETCOUNT	5237	
		1D	1B	004E2	BLEQU	43\$	:			
	7E	00000000'	EF	9A	004E4	MOVZBL	COM_O_SETCOUNT, -(SP)	5241		
	7E	00000000'	EF	9A	004EB	MOVZBL	QUAC+79, -(SP)			
		02	DD	004F2	PUSHL	#2	:	5239		
	00000000G	8F	DD	004F4	PUSHL	#BACKUP\$ NOSUCHRVN	:			
	00000000G	00	04	FB	004FA	CALLS	#4, LIB\$SIGNAL			
OC	20	00000000'	FF	2C	00501	43\$:	MOVCS	OUTPUT_ATTBUF+76, @OUTPUT_ATTBUF+80, #32, -	5243	
		00000000'	EF		0050E					
00000000'	EF	04	AE	8F	28	00513	44\$:	MOVCS	#144, SAVE_ATTBUF, OUTPUT_ATTBUF	5254

			57	04	0051E	RET		4891	
			01	DD	0051F	PUSHL	R7	5264	
4F	00000000G	00	03	FB	00521	CALLS	#1, GET_VOLUME_ATTRIBUTES		
	00000000'	EF	05	E1	00528	BBC	#3, QUAL+10, 48\$	5266	
			56	95	00530	TSTB	QUAL+8	5267	
			56	19	00536	BLSS	50\$		
	22	00000000'	EF	E9	00538	BLBC	QUAL+14, 46\$	5268	
	50	00000000'	EF	9A	0053F	MOVZBL	QUAL+79, R0	5269	
	00000000'	EF	50	B1	00546	CMPL	R0, OUTPUT_ATTBUF+66		
			12	13	0054D	BEQL	46\$		
	01	00000000'	EF	91	0054F	CMPL	QUAL+79, #1	5270	
			57	12	00556	BNEQ	52\$		
	01	00000000'	EF	B1	00558	CMPL	OUTPUT_ATTBUF+66, #1		
			58	1A	0055F	BGTRU	54\$		
	00000000G	00	00	FB	00561	CALLS	#0, STA_INIVOL	5272	
			04	04	00568	RET		4891	
	0101	8F	10	A7	B1	00569	CMPL	16(R7), #257	5283
			0E	13	0056F	BEQL	48\$		
			8F	DD	00571	PUSHL	#BACKUP\$ INVATTSTR	5285	
	00000000G	00	01	FB	00577	CALLS	#1, LIB\$SIGNAL		
			04	04	0057E	RET			
01	00000000'	EF	03	E0	0057F	BBS	#3, QUAL+10, 49\$	5288	
			04	04	00587	RET			
			EF	95	00588	TSTB	QUAL+8	5289	
			01	18	0058E	BGEQ	51\$		
			04	04	00590	RET			
			EF	E9	00591	BLBC	QUAL+14, 55\$	5290	
	24	00000000'	EF	9A	00598	MOVZBL	QUAL+79, R0	5291	
	50	00000000'	EF	B1	0059F	CMPL	R0, OUTPUT_ATTBUF+66		
	00000000'	EF	50	B1	0059F	CMPL	R0, OUTPUT_ATTBUF+66		
			14	13	005A6	BEQL	55\$		
	01	00000000'	EF	91	005A8	CMPL	QUAL+79, #1	5292	
			01	13	005AF	BEQL	53\$		
			04	04	005B1	RET			
	01	00000000'	EF	B1	005B2	CMPL	OUTPUT_ATTBUF+66, #1		
			01	1B	005B9	BLEQU	55\$		
			04	04	005BB	RET			
			A7	9F	005BC	PUSHAB	16(R7)	5294	
	00000000G	00	01	FB	005BF	CALLS	#1, STA_INIT_HDRS		
			04	04	005C6	RET		4891	
			57	DD	005C7	PUSHL	R7	5304	
	00000000G	00	01	FB	005C9	CALLS	#1, GET_PHYSVOL_ATTRIBUTES		
01	00000000'	EF	03	E1	005D0	BBC	#3, COM_FLAGS, 57\$	5305	
			04	04	005D8	RET			
07	04	A7	02	E1	005D9	BBC	#2, 4(R7), 58\$	5310	
	00000000'	EF	01	88	005DE	BISB2	#1, OUTPUT_FLAGS+1	5312	
	00000000'	EF	EF	91	005E5	CMPL	OUTPUT_ATTBUF+34, OUTPUT_DEVGEOM	5317	
			27	12	005F0	BNEQ	59\$		
	00000000'	EF	EF	91	005F2	CMPL	OUTPUT_ATTBUF+35, OUTPUT_DEVGEOM+1	5318	
			1A	12	005FD	BNEQ	59\$		
	00000000'	EF	EF	B1	005FF	CMPL	OUTPUT_ATTBUF+32, OUTPUT_DEVGEOM+2	5319	
			0D	12	0060A	BNEQ	59\$		
	00000000'	EF	EF	D1	0060C	CMPL	OUTPUT_ATTBUF+24, OUTPUT_DEVGEOM+4	5320	
			13	13	00617	BEQL	60\$		
			EF	DD	00619	PUSHL	OUTPUT_FAB	5322	
			8F	DD	0061F	PUSHL	#BACKUP\$ PHYDEVGEOM		
	00000000G	00	02	FB	00625	CALLS	#2, FILE_ERROR		
	59	00000000'	FF	D0	0062C	MOVL	@OUTPUT_BAD, R9	5330	

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

G 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 132
(20)

	54	01	CE	00633	MNEGL	#1, I			
		64	11	00636	BRB	68\$			
52	00000000'	08	C1	00638	61\$: ADDL3	#8, OUTPUT_BAD, Q	5333		
50		62	A2	00640	ADDL3	4(Q), (Q), R0	5334		
		58	FF	00645	MOVAB	-1(R0), R8			
		53	62	00649	MOVL	(Q), LBN			
		46	11	0064C	BRB	67\$			
	50 00000000'	EF	D0	0064E	62\$: MOVL	OUTPUT_ATTBUF+20, P	5337		
	55 00000000'	EF	3C	00655	MOVZWL	OUTPUT_ATTBUF+16, R5	5342		
	55 00000000'	EF	C0	0065C	ADDL2	OUTPUT_ATTBUF+20, R5			
		50	D1	00663	63\$: CMPL	P, R5	5341		
		16	1E	00666	BGEQU	65\$			
	60	53	D1	00668	CMPL	LBN, (P)	5346		
		0C	1F	0066B	BLSSU	64\$			
51		60	04	A0	C1	0066D	ADDL3	4(P), (P), R1	5347
		51	D7	00672	DECL	R1			
		53	D1	00674	CMPL	LBN, R1			
		19	1B	00677	BLEQU	66\$			
		50	08	C0	00679	64\$: ADDL2	#8, P	5350	
			E5	11	0067C	BRB	63\$	5340	
			EF	DD	0067E	65\$: PUSHL	OUTPUT_FAB	5355	
			8F	DD	00684	PUSHL	#BACKUP\$_BADBLOCK		
00000000G	00	02	FB	0068A	CALLS	#2, FILE_ERROR			
			04	00691	RET			5354	
		53	D6	00692	66\$: INCL	LBN		5334	
		53	D1	00694	67\$: CMPL	LBN, R8			
		58	B5	1B	00697	BLEQU	62\$		
		52	08	C0	00699	ADDL2	#8, Q	5359	
98		54	59	F2	0069C	68\$: AOBLSS	R9, I, 61\$	5330	
			04	006A0	RET			4891	
	E4CD	CF	00	FB	006A1	69\$: CALLS	#0, CLOSE_RESTORE	5372	
			57	DD	006A6	PUSHL	R7	5377	
	00000000G	00	01	FB	006A8	CALLS	#1, GET_FILE_ATTRIBUTES		
33	00000000'	EF	06	E1	006AF	BBC	#6, QUAC+13, -72\$	5379	
	F0	8F	00000000'	EF	93	006B7	BITB	OUTPUT_RECATTR, #240	5380
			29	12	006BF	BNEQ	72\$		
50	00000000'	EF	10	9C	006C1	ROTL	#16, OUTPUT_RECATTR+8, EOF_SIZE	5386	
			EF	B5	006C9	TSTW	OUTPUT_RECATTR+12	5387	
			02	12	006CF	BNEQ	70\$		
			50	D7	006D1	DECL	EOF_SIZE	5388	
	00000000'	EF	50	D1	006D3	70\$: CMPL	R0, OUTPUT_ATTBUF+92	5390	
			07	1B	006DA	BLEQU	71\$		
			EF	D0	006DC	MOVL	OUTPUT_ATTBUF+92, R0		
	50 00000000'	EF	50	D0	006E3	71\$: MOVL	R0, OUTPUT_ATTBUF+92		
	50 00000000'	EF	D0	006EA	72\$: MOVL	OUTPUT_BCB, R0		5392	
	50 00000000'	EF	D0	006F1	MOVL	12(R0), BUFFER			
	50 00000000'	EF	D0	006F5	MOVL	OUTPUT_ATTBUF+80, 80(BUFFER)		5393	
	54 00000000'	EF	B0	006FD	MOVW	OUTPUT_ATTBUF+84, 84(BUFFER)		5395	
0F	00000000'	EF	03	E1	00705	BBC	#3, COM_FLAGS, 73\$	5401	
			EF	95	0070D	TSTB	QUAL+15		
			07	18	00713	BGEQ	73\$		
			01	DD	00715	PUSHL	#1	5402	
	DE32	CF	01	FB	00717	CALLS	#1, ASSIGN_VERIFY_CHANNEL		
	EC7E	CF	00	FB	0071C	73\$: CALLS	#0, INITIACIZE_AND_SELECT	5408	
		03	50	E8	00721	BLBS	R0, 74\$		
			0206	31	00724	BRW	94\$		
39	04	A7	01	E1	00727	74\$: BBC	#1, 4(R7), 77\$	5411	

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

H 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 133
(20)

0040 8F

08

		00000000'	EF	95	0072C	TSTB	QUAL+8	5413
				22	18 00732	BGEQ	76\$	
08	00000000'	EF		05	E1 00734	BBC	#5, QUAL+15, 75\$	5416
31	00000000'	EF		03	E0 0073C	BBS	#3, QUAL+10, 78\$	
2E	00000000'	EF		03	E1 00744	BBC	#3, COM_FLAGS, 79\$	5417
		00000000'	EF	95	0074C	TSTB	QUAL+15	
				26	18 00752	BGEQ	79\$	
				1F	11 00754	BRB	78\$	5419
17	00000000'	EF		03	E0 00756	BBS	#3, QUAL+10, 78\$	5423
	E94D	CF		00	FB 0075E	CALLS	#0, CREATE_DIRECTORY	5427
				15	11 00763	BRB	79\$	5413
08	00000000'	EF		04	E1 00765	BBC	#4, QUAL+10, 78\$	5431
	EE7A	CF		00	FB 0076D	CALLS	#0, SELECT_INCREMENTAL	5433
		OS		50	E9 00772	BLBC	R0, 79\$	
	F23C	CF		00	FB 00775	CALLS	#0, CREATE_OR_OPEN_FILE	5436
00		6E		00	2C 0077A	MOVCS	#0, (SP), #0, #64, ACL_FIB	5441
		00000000'	EF		00781			
	00000000'	EF		20	88 00786	BISB2	#32, ACL_FIB+2	5442
18	00000000'	EF		03	E1 0078D	BBC	#3, QUAL+10, 80\$	5443
	00000000'	EF	00000000'	EF	D0 00795	MOVL	OUTPUT_ATTBUF+80, ACL_FIB+4	5446
	00000000'	EF	00000000'	EF	B0 007A0	MOVW	OUTPUT_ATTBUF+84, ACL_FIB+8	5448
				17	11 007AB	BRB	81\$	5443
		50	00000000'	EF	D0 007AD	MOVL	OUTPUT_NAM, R0	5452
	00000000'	FF	24	A0	D0 007B4	MOVL	36(R0), ACL_FIB+4	
	00000000'	EF	28	A0	B0 007BC	MOVW	40(R0), ACL_FIB+8	5454
	00000000'	EF		20	8A 007C4	BICB2	#32, OUTPUT_FLAGS	5456
	00000000'	EF	40	8F	88 007CB	BISB2	#64, OUTPUT_FLAGS	5457
00		6E		00	2C 007D3	MOVCS	#0, (SP), #0, #8, ACL_FIB_DESCR	5458
		00000000'	EF		007D8			
	00000000'	EF	40	8F	9B 007DD	MOVZBW	#64, ACL_FIB_DESCR	5459
	00000000'	EF	00000000'	EF	9E 007E5	MOVAB	ACL_FIB, ACL_FIB_DESCR+4	5460
			00000000'	EF	B5 007F0	TSTW	OUTPUT_ACLSEGMENT	5465
				03	12 007F6	BNEQ	83\$	
				01	32 31 007F8	BRW	94\$	
F5	00000000'	EF		03	E0 007FB	BBS	#3, COM_FLAGS, 82\$	5466
		10	00000000'	EF	E8 00803	BLBS	OUTPUT_FLAGS, 84\$	5467
08	00000000'	EF		03	E0 0080A	BBS	#3, OUTPUT_FLAGS, 84\$	
			00000000'	EF	95 00812	TSTB	OUTPUT_FLAGS	5468
				DE	18 00818	BGEQ	82\$	
D6	00000000'	EF		01	E0 0081A	BBS	#1, QUAL+14, 82\$	5469
79	00000000'	EF		06	E1 00822	BBC	#6, OUTPUT_FLAGS, 87\$	5482
	00000000'	EF	40	8F	8A 0082A	BICB2	#64, OUTPUT_FLAGS	5485
	F4	AD	00240008	8F	D0 00832	MOVL	#2359304, ATR_DESC	5486
	F8	AD	78	AE	9E 0083A	MOVAB	DUMMY_ACL, ATR_DESC+4	5488
			FC	AD	D4 0083F	CLRL	ATR_DESC+8	5489
		50	00000000'	EF	D0 00842	MOVL	OUTPUT_CHAN, R0	5495
0001FFFF		8F		50	D1 00849	CMPL	R0, #131071	
				23	1F 00850	BLSSU	85\$	
				7E	D4 00852	CLRL	-(SP)	
		F4		AD	9F 00854	PUSHAB	ATR_DESC	
				7E	7C 00857	CLRQ	-(SP)	
				7E	D4 00859	CLRL	-(SP)	
		00000000'	EF	9F	0085B	PUSHAB	ACL_FIB_DESCR	
				7E	7C 00861	CLRQ	-(SP)	
		EC		AD	9F 00863	PUSHAB	IOSB	
				36	DD 00866	PUSHL	#54	
				50	DD 00868	PUSHL	R0	

00000000G	00	7E	D4	0086A	CLRL	-(SP)	
		0C	FB	0086C	CALLS	#12, STA_QIOW	
		21	11	00873	BRB	86\$	
		7E	D4	00875	85\$:	CLRL	-(SP)
	F4	AD	9F	00877	PUSHAB	ATR_DESC	
		7E	7C	0087A	CLRQ	-(SP)	
		7E	D4	0087C	CLRL	-(SP)	
	00000000'	EF	9F	0087E	PUSHAB	ACL_FIB_DESCR	
		7E	7C	00884	CLRQ	-(SP)	
	EC	AD	9F	00886	PUSHAB	IOSB	
		36	DD	00889	PUSHL	#54	
		50	DD	0088B	PUSHL	R0	
		7E	D4	0088D	CLRL	-(SP)	
00000000G	9F	0C	FB	0088F	CALLS	#12, @#SYSSQIOW	
	52	50	D0	00896	86\$:	MOVL	R0, STATUS
	7E	52	E9	00899	BLBC	STATUS, 91\$	5496
	52	EC	AD	3C	0089C	MOVZWL	IOSB, STATUS
	7E	52	E9	008A0	BLBC	STATUS, 92\$	5497
01 00000000'	EF	05	E1	008A3	87\$:	BBC	#5, OUTPUT_FLAGS, 88\$
			04	008AB	RET		5512
	F4	AD	00000000'	EF	B0	008AC	88\$:
	F6	AD		1F	B0	008B4	MOVW
	F8	AD	00000000'	EF	D0	008B8	MOVW
		FC	AD	D4	008C0	CLRL	ATR_DESC+8
	50	00000000'	EF	D0	008C3	MOVL	OUTPUT_CHAN, R0
0001FFFF	8F	50	D1	008CA	CMPL	R0, #131071	5519
		23	1F	008D1	BLSSU	89\$	5524
		7E	D4	008D3	CLRL	-(SP)	
	F4	AD	9F	008D5	PUSHAB	ATR_DESC	
		7E	7C	008D8	CLRQ	-(SP)	
		7E	D4	008DA	CLRL	-(SP)	
	00000000'	EF	9F	008DC	PUSHAB	ACL_FIB_DESCR	
		7E	7C	008E2	CLRQ	-(SP)	
	EC	AD	9F	008E4	PUSHAB	IOSB	
		36	DD	008E7	PUSHL	#54	
		50	DD	008E9	PUSHL	R0	
		7E	D4	008EB	CLRL	-(SP)	
00000000G	00	0C	FB	008ED	CALLS	#12, STA_QIOW	
		21	11	008F4	BRB	90\$	
		7E	D4	008F6	89\$:	CLRL	-(SP)
	F4	AD	9F	008F8	PUSHAB	ATR_DESC	
		7E	7C	008FB	CLRQ	-(SP)	
		7E	D4	008FD	CLRL	-(SP)	
	00000000'	EF	9F	008FF	PUSHAB	ACL_FIB_DESCR	
		7E	7C	00905	CLRQ	-(SP)	
	EC	AD	9F	00907	PUSHAB	IOSB	
		36	DD	0090A	PUSHL	#54	
		50	DD	0090C	PUSHL	R0	
		7E	D4	0090E	CLRL	-(SP)	
00000000G	9F	0C	FB	00910	CALLS	#12, @#SYSSQIOW	
	52	50	D0	00917	90\$:	MOVL	R0, STATUS
	04	52	E9	0091A	91\$:	BLBC	STATUS, 92\$
	52	EC	AD	3C	0091D	MOVZWL	IOSB, STATUS
	03	52	E8	00921	92\$:	BLBS	STATUS, 93\$
		014C	31	00924	93\$:	BRW	107\$
	00000000'	EF	B4	00927	94\$:	CLRW	OUTPUT_ACLSEGMENT
00000000'	EF	02	8A	0092D	BICB2	#2, OUTPUT_FLAGS	5536

RESTORE
V04-000

Restore a save set
RESTORE_ONE_RECORD - restore save set record

J 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 135
(20)

00000000G	00	00000000'	57	04	00934	95\$:	RET		4891
			01	DD	00935		PUSHL	R7	5553
			01	FB	00937		CALLS	#1, GET_FILE_EXT_ATTRIBUTES	
			EF	B5	0093E		TSTW	OUTPUT_ACLSEGMENT	5558
			01	12	00944		BNEQ	96\$	
				04	00946		RET		
01	00000000'	EF	03	E1	00947	96\$:	BBC	#3, COM_FLAGS, 97\$	5559
				04	0094F		RET		
09	00000000'	11	EF	E8	00950	97\$:	BLBS	OUTPUT_FLAGS, 98\$	5560
		EF	03	E0	00957		BBS	#3, OUTPUT_FLAGS, 98\$	
		00000000'	EF	95	0095F		TSTB	OUTPUT_FLAGS	5561
			01	19	00965		BLSS	98\$	
				04	00967		RET		
01	00000000'	EF	01	E1	00968	98\$:	BBC	#1, QUAL+14, 99\$	5562
				04	00970		RET		
79	00000000'	EF	06	E1	00971	99\$:	BBC	#6, OUTPUT_FLAGS, 102\$	5575
	00000000'	EF	8F	8A	00979		BICB2	#6, OUTPUT_FLAGS	5578
	F4	AD	8F	D0	00981		MOVL	#2359304, ATR_DESC	5579
	F8	AD	AE	9E	00989		MOVAB	DUMMY_ACL, ATR_DESC+4	5581
			AD	D4	0098E		CLRL	ATR_DESC+8	5582
		50	EF	D0	00991		MOVL	OUTPUT_CHAN, R0	5588
0001FFFF		8F	50	D1	00998		CMPL	R0, #131071	
			23	1F	0099F		BLSSU	100\$	
			7E	D4	009A1		CLRL	-(SP)	
		F4	AD	9F	009A3		PUSHAB	ATR_DESC	
			7E	7C	009A6		CLRQ	-(SP)	
			7E	D4	009A8		CLRL	-(SP)	
		00000000'	EF	9F	009AA		PUSHAB	ACL_FIB_DESCR	
			7E	7C	009B0		CLRQ	-(SP)	
		EC	AD	9F	009B2		PUSHAB	IOSB	
			36	DD	009B5		PUSHL	#54	
			50	DD	009B7		PUSHL	R0	
			7E	D4	009B9		CLRL	-(SP)	
00000000G	00		0C	FB	009BB		CALLS	#12, STA_Q10W	
			21	11	009C2		BRB	101\$	
			7E	D4	009C4	100\$:	CLRL	-(SP)	
		F4	AD	9F	009C6		PUSHAB	ATR_DESC	
			7E	7C	009C9		CLRQ	-(SP)	
			7E	D4	009CB		CLRL	-(SP)	
		00000000'	EF	9F	009CD		PUSHAB	ACL_FIB_DESCR	
			7E	7C	009D3		CLRQ	-(SP)	
		EC	AD	9F	009D5		PUSHAB	IOSB	
			36	DD	009D8		PUSHL	#54	
			50	DD	009DA		PUSHL	R0	
			7E	D4	009DC		CLRL	-(SP)	
00000000G	9F		0C	FB	009DE		CALLS	#12, @#SYSSQ10W	
	52		50	D0	009E5	101\$:	MOVL	R0, STATUS	
	7E		52	E9	009E8		BLBC	STATUS, 106\$	5589
	52	EC	AD	3C	009EB		MOVZWL	IOSB, STATUS	
	77		52	E9	009EF		BLBC	STATUS, 106\$	5590
01	00000000'	EF	05	E1	009F2	102\$:	BBC	#5, OUTPUT_FLAGS, 103\$	5605
				04	009FA		RET		
	F4	AD	EF	B0	009FB	103\$:	MOVW	OUTPUT_ACLSEGMENT, ATR_DESC	5609
	F6	AD	1F	B0	00A03		MOVW	#31, ATR_DESC+2	5610
	F8	AD	EF	D0	00A07		MOVL	OUTPUT_ACLSEGMENT+4, ATR_DESC+4	5611
			AD	D4	00A0F		CLRL	ATR_DESC+8	5612
		50	EF	D0	00A12		MOVL	OUTPUT_CHAN, R0	5617

Restore a save set
RESTORE_ONE_RECORD - restore save set record

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

(20)

5618
5619
5625
5624
5623

5626
5621
5629
5635

; Routine Size: 2711 bytes, Routine Base: CODE + 1CB9

RESTORE
V04-000

Restore a save set
RESTORE_VERIFY_REEL - verify save set volume

L 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 137
(21)

```

: 4102 5636 1 %SBTTL 'RESTORE_VERIFY_REEL - verify save set volume'
: 4103 5637 1 GLOBAL ROUTINE RESTORE_VERIFY_REEL: NOVALUE=
: 4104 5638 1
: 4105 5639 1 ++
: 4106 5640 1
: 4107 5641 1 FUNCTIONAL DESCRIPTION:
: 4108 5642 1 This routine is called at the end of each reel to verify that reel.
: 4109 5643 1
: 4110 5644 1 INPUT PARAMETERS:
: 4111 5645 1 NONE
: 4112 5646 1
: 4113 5647 1 IMPLICIT INPUTS:
: 4114 5648 1 NONE
: 4115 5649 1
: 4116 5650 1 OUTPUT PARAMETERS:
: 4117 5651 1 NONE
: 4118 5652 1
: 4119 5653 1 IMPLICIT OUTPUTS:
: 4120 5654 1 NONE
: 4121 5655 1
: 4122 5656 1 ROUTINE VALUE:
: 4123 5657 1 NONE
: 4124 5658 1
: 4125 5659 1 SIDE EFFECTS:
: 4126 5660 1 NONE
: 4127 5661 1
: 4128 5662 1 --
: 4129 5663 1
: 4130 5664 2 BEGIN
: 4131 5665 2 LOCAL
: 4132 5666 2 STATUS, | Status variable
: 4133 5667 2 IOSB: VECTOR[4,WORD], | I/O status block
: 4134 5668 2 FIB: BBLOCK[10], | FIB
: 4135 5669 2 FIB_DESC: VECTOR[2], | Descriptor for FIB
: 4136 5670 2 SAVE_OUTPUT_BAD, | Save for OUTPUT_BAD
: 4137 5671 2 LOCAL_SAVE: BBLOCK[OUTPUT_END-OUTPUT_BEG]; | Save area
: 4138 5672 2
: 4139 5673 2
: 4140 5674 2 | If a file is accessed, deaccess it for the duration of the verify pass to
: 4141 5675 2 | avoid having the file open for the entire time.
: 4142 5676 2
: 4143 5677 2 IF .OUTPUT_FLAGS[OUTPUT_OPEN] AND NOT .QUAL[QUAL_PHYS]
: 4144 5678 2 THEN
: 4145 5679 3 BEGIN
: 4146 5680 3 STATUS = C$QIOW(
: 4147 5681 3 FUNC=IOS DEACCESS,
: 4148 5682 3 CHAN=.OUTPUT_CHAN,
: 4149 5683 3 IOSB=IOSB);
: 4150 5684 3 IF .STATUS THEN STATUS = .IOSB[0];
: 4151 5685 3 IF NOT .STATUS
: 4152 5686 3 THEN
: 4153 5687 3 FILE ERROR(
: 4154 5688 3 BACKUPS_CLOSEOUT + ST$K_ERROR,
: 4155 5689 3 .OUTPUT_FAB,
: 4156 5690 3 .STATUS);
: 4157 5691 2 END;
: 4158 5692 2
```

RESTORE
V04-000

Restore a save set
RESTORE_VERIFY_REEL - verify save set volume

M 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 138
(21)

```

: 4159 5693 2
: 4160 5694 2 ! Readjust context.
: 4161 5695 2
: 4162 5696 2 QUAL[QUAL_COMP] = TRUE;
: 4163 5697 2 IF .QUAL[QUAL_PHYS] THEN VERIFY FAB = .OUTPUT FAB;
: 4164 5698 2 CH$MOVE(OUTPUT_END-OUTPUT_BEG, OUTPUT_BEG, LOCAL_SAVE);
: 4165 5699 2 SAVE OUTPUT_BAD = .OUTPUT_BAD;
: 4166 5700 2 CH$FILL(0, OUTPUT_END-OUTPUT_BEG, OUTPUT_BEG);
: 4167 5701 2 OUTPUT_BAD = .SAVE_OUTPUT_BAD;
: 4168 5702 2
: 4169 5703 2
: 4170 5704 2 ! Do the verify by calling the RESTORE module.
: 4171 5705 2
: 4172 5706 2 RESTORE();
: 4173 5707 2
: 4174 5708 2
: 4175 5709 2 ! Restore context.
: 4176 5710 2
: 4177 5711 2 QUAL[QUAL_COMP] = FALSE;
: 4178 5712 2 CH$MOVE(OUTPUT_END-OUTPUT_BEG, LOCAL_SAVE, OUTPUT_BEG);
: 4179 5713 2
: 4180 5714 2
: 4181 5715 2 ! If necessary, re-access the file that was accessed at the end of the
: 4182 5716 2 ! previous reel.
: 4183 5717 2
: 4184 5718 2 IF .OUTPUT_FLAGS[OUTPUT_OPEN] AND NOT .QUAL[QUAL_PHYS]
: 4185 5719 2 THEN
: 4186 5720 3 BEGIN
: 4187 5721 3 FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NOWRITE OR FIB$M_NORECORD;
: 4188 5722 3 FIB[FIB$W_FID_NUM] = .OUTPUT_NAM[NAM$W_FID_NUM];
: 4189 5723 3 FIB[FIB$W_FID_SEQ] = .OUTPUT_NAM[NAM$W_FID_SEQ];
: 4190 5724 3 FIB[FIB$W_FID_RVN] = .OUTPUT_NAM[NAM$W_FID_RVN];
: 4191 5725 3 FIB_DESC[0] = 10;
: 4192 5726 3 FIB_DESC[1] = FIB;
: 4193 5727 3 STATUS = C$QIOW(
P 5728 3 FUNC=IOS_ACCESS OR IOSM_ACCESS,
P 5729 3 CHAN=.OUTPUT_CHAN,
P 5730 3 IOSB=IOSB,
: 4197 5731 3 P1=FIB_DESC);
: 4198 5732 3 IF .STATUS THEN STATUS = .IOSB[0];
: 4199 5733 3 IF NOT .STATUS
: 4200 5734 3 THEN
: 4201 5735 4 BEGIN
: 4202 5736 4 FILE_ERROR(
: 4203 5737 4 BACKUP$CONTINUE,
: 4204 5738 4 .OUTPUT_FAB,
: 4205 5739 4 .STATUS);
: 4206 5740 4 CLOSE_RESTORE();
: 4207 5741 3 END;
: 4208 5742 2 END;
: 4209 5743 1 END;
```

OFFC 00000

.ENTRY RESTORE_VERIFY_REEL, Save R2,R3,R4,R5,R6,- ; 5637

RESTORE
V04-000

Restore a save set
RESTORE_VERIFY_REEL - verify save set volume

N 14
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 139
(21)

		5B	00000000G	00	9E	00002	MOVAB	R7,R8,R9,R10,R11	
		5A	00000000G	9F	9E	00009	MOVAB	FILE ERROR, R11	
		59	00000000G	00	9E	00010	MOVAB	#SYSSQIOW, R10	
		58	00000000'	EF	9E	00017	MOVAB	STA QIOW, R9	
		5E	FF28	CE	9E	0001E	MOVAB	OUTPUT CHAN, R8	
		57	04	A8	E9	00023	MOVAB	-216(SP), SP	
51	FEBO	C8		05	E0	00027	BLBC	OUTPUT_FLAGS, 4\$	5677
	0001FFFF	50		68	D0	0002D	BBS	#5, QUAL+12, 4\$	
		8F		50	D1	00030	MOVL	OUTPUT CHAN, R0	5683
				16	1F	00037	CMPL	R0, #131071	
				7E	7C	00039	BLSSU	1\$	
				7E	7C	0003B	CLRQ	-(SP)	
				7E	7C	0003D	CLRQ	-(SP)	
				7E	7C	0003F	CLRQ	-(SP)	
			F8	AD	9F	00041	CLRQ	-(SP)	
				34	DD	00044	PUSHAB	IOSB	
				50	DD	00046	PUSHL	#52	
				7E	D4	00048	PUSHL	R0	
		69		0C	FB	0004A	CLRL	-(SP)	
				14	11	0004D	CALLS	#12, STA_QIOW	
				7E	7C	0004F	BRB	2\$	
				7E	7C	00051	CLRQ	-(SP)	
				7E	7C	00053	CLRQ	-(SP)	
				7E	7C	00055	CLRQ	-(SP)	
			F8	AD	9F	00057	CLRQ	-(SP)	
				34	DD	0005A	PUSHAB	IOSB	
				50	DD	0005C	PUSHL	#52	
				7E	D4	0005E	PUSHL	R0	
		6A		0C	FB	00060	CLRL	-(SP)	
		57		50	D0	00063	CALLS	#12, SYSSQIOW	
		07		57	E9	00066	MOVL	R0, STATUS	
		57		AD	3C	00069	BLBC	STATUS, 3\$	5684
		0E		57	E8	0006D	MOVZWL	IOSB, STATUS	
				57	DD	00070	BLBS	STATUS, 4\$	5685
			08	A8	DD	00072	PUSHL	STATUS	5690
			00000000G	8F	DD	00075	PUSHL	OUTPUT FAB	5689
		6B		03	FB	0007B	PUSHL	#BACKUP\$_CLOSEOUT+2	5688
		C8		8F	88	0007E	CALLS	#3, FILE_ERROR	
06	FEAC	C8	80	05	E1	00084	BISB2	#128, QUAL+8	5696
	FEBO	C8		A8	D0	0008A	BBC	#5, QUAL+12, 5\$	5697
	00C4	C8	08	8F	28	00090	MOVL	OUTPUT FAB, VERIFY FAB	
6E		68	00BC	8F	28	00090	MOV3	#188, OUTPUT_BEG, LOCAL_SAVE	5698
		56	18	A8	D0	00096	MOVL	OUTPUT_BAD, SAVE OUTPUT_BAD	5699
00BC		6E		00	2C	0009A	MOV5	#0, (SP), #0, #188, OUTPUT_BEG	5700
				68		000A1			
				56	D0	000A2	MOVL	SAVE OUTPUT_BAD, OUTPUT_BAD	5701
		18		00	FB	000A6	CALLS	#0, RESTORE	5706
	0000V	CF		8F	8A	000AB	BICB2	#128, QUAL+8	5711
	FEAC	C8	80	8F	28	000B1	MOV3	#188, LOCAL_SAVE, OUTPUT_BEG	5712
68		6E	00BC	A8	E8	000B7	BLBS	OUTPUT_FLAGS, 6\$	5718
		01	04		04	000BB	RET		
				05	E0	000BC	BBS	#5, QUAL+12, 10\$	
7F	FEBO	C8	00200101	8F	DC	000C2	MOVL	#2097409, FIB	5721
	EC	50	0C	A8	D0	000CA	MOVL	OUTPUT_NAM, R0	5722
				A0	D0	000CE	MOVL	36(R0), FIB+4	
	F0	AD	24	A0	B0	000D3	MOVW	40(R0), FIB+8	5724
	F4	AD	28	0A	D0	000D8	MOVL	#10, FIB_DESC	5725
	E4	AD							

RESTORE
V04-000

Restore a save set
RESTORE_VERIFY_REEL - verify save set volume

B 15

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742

[BACKUP.SRC]RESTORE.B32;1

Page 140

(21)

E8	AD	EC	AD	9E	000DC	MOVAB	FIB, FIB_DESC+4		5726
	50		68	D0	000E1	MOVL	OUTPUT_CHAN, R0		5731
0001FFFF	8F		50	D1	000E4	CMPL	R0, #131071		
			1B	1F	000EB	BLSSU	7\$		
			7E	7C	000FD	CLRQ	-(SP)		
			7E	7C	0CCE	CLRQ	-(SP)		
			7E	D4	000F1	CLRL	-(SP)		
		E4	AD	9F	000F3	PUSHAB	FIB_DESC		
			7E	7C	000F6	CLRQ	-(SP)		
		F8	AD	9F	000F8	PUSHAB	IOSB		
	7E	72	8F	9A	000FB	MOVZBL	#114, -(SP)		
			50	DD	000FF	PUSHL	R0		
			7E	D4	00101	CLRL	-(SP)		
	69		0C	FB	00103	CALLS	#12, STA_QIOW		
			19	11	00106	BRB	8\$		
			7E	7C	00108	CLRQ	-(SP)		
			7E	7C	0010A	CLRQ	-(SP)		
			7E	D4	0010C	CLRL	-(SP)		
		E4	AD	9F	0010E	PUSHAB	FIB_DESC		
			7E	7C	00111	CLRQ	-(SP)		
		F8	AD	9F	00113	PUSHAB	IOSB		
	7E	72	8F	9A	00116	MOVZBL	#114, -(SP)		
			50	DD	0011A	PUSHL	R0		
			7E	D4	0011C	CLRL	-(SP)		
	6A		0C	FB	0011E	CALLS	#12, SYSSQIOW		
	57		50	D0	00121	MOVL	R0, STATUS		
	07		57	E9	00124	BLBC	STATUS, 9\$		5732
	57	F8	AD	3C	00127	MOVZWL	IOSB, STATUS		
	13		57	E8	0012B	BLBS	STATUS, 10\$		5733
			57	DD	0012E	PUSHL	STATUS		5739
		08	A8	DD	00130	PUSHL	OUTPUT_FAB		5738
		00000000G	8F	DD	00133	PUSHL	#BACKUP\$-CONTINUE		5736
	6B		03	FB	00139	CALLS	#3, FILE_ERROR		
DF9B	CF		00	FB	0013C	CALLS	#0, CLOSE_RESTORE		5740
			04	00141	10\$:	RET			5743

; Routine Size: 322 bytes, Routine Base: CODE + 2750

RESTORE
V04-000

Restore a save set
INIT_RESTORE - initialize restore operation

C 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 141
(22)

```
: 4211 5744 1 %SBTTL 'INIT RESTORE - initialize restore operation'
: 4212 5745 1 GLOBAL ROUTINE INIT_RESTORE: NOVALUE=
: 4213 5746 1
: 4214 5747 1 !++
: 4215 5748 1
: 4216 5749 1 FUNCTIONAL DESCRIPTION:
: 4217 5750 1 This routine initializes the restore module.
: 4218 5751 1
: 4219 5752 1 INPUT PARAMETERS:
: 4220 5753 1 NONE
: 4221 5754 1
: 4222 5755 1 IMPLICIT INPUTS:
: 4223 5756 1 NONE
: 4224 5757 1
: 4225 5758 1 OUTPUT PARAMETERS:
: 4226 5759 1 NONE
: 4227 5760 1
: 4228 5761 1 IMPLICIT OUTPUTS:
: 4229 5762 1 OWN storage initialized.
: 4230 5763 1
: 4231 5764 1 ROUTINE VALUE:
: 4232 5765 1 NONE
: 4233 5766 1
: 4234 5767 1 SIDE EFFECTS:
: 4235 5768 1 NONE
: 4236 5769 1
: 4237 5770 1 !--
: 4238 5771 1
: 4239 5772 2 BEGIN
: 4240 5773 2
: 4241 5774 2 ! Find the output qualifiers block.
: 4242 5775 2
: 4243 5776 2 OUTPUT_QUAL = .QUAL[QUAL_OUTP_LIST];
: 4244 5777 2
: 4245 5778 2
: 4246 5779 2 IF .COM_FLAGS[COM_VERIFYING]
: 4247 5780 2 THEN
: 4248 5781 3 BEGIN
: 4249 5782 3
: 4250 5783 3 ! This is a verify pass. Use the verification FAB as if it were the output
: 4251 5784 3 ! FAB during the verify pass.
: 4252 5785 3
: 4253 5786 3 OUTPUT_FAB = .VERIFY_FAB;
: 4254 5787 3 IF .QUAL[QUAL_OSAV]
: 4255 5788 3 THEN
: 4256 5789 3 ASSIGN_VERIFY_CHANNEL(FALSE)
: 4257 5790 3 ELSE
: 4258 5791 4 BEGIN
: 4259 5792 4 LOCAL
: 4260 5793 4 FAB: REF BBLOCK; ! Pointer to real output FAB
: 4261 5794 4
: 4262 5795 4 FAB = .OUTPUT_QUAL[QUAL_PARA_FC];
: 4263 5796 4 OUTPUT_FAB[FAB$B_FNS] = .FAB[FAB$B_FNS];
: 4264 5797 4 OUTPUT_FAB[FAB$B_FNA] = .FAB[FAB$B_FNA];
: 4265 5798 4 BBLOCK[OUTPUT_FAB[FC_NAM], NAM$B_ESL] = .BBLOCK[FAB[FC_NAM], NAM$B_ESL];
: 4266 5799 4 BBLOCK[OUTPUT_FAB[FC_NAM], NAM$B_ESA] = .BBLOCK[FAB[FC_NAM], NAM$B_ESA];
: 4267 5800 3 END;
```

```

: 4268 5801 3      END
: 4269 5802 2      ELSE
: 4270 5803 3      BEGIN
: 4271 5804 3          OUTPUT_FAB = .OUTPUT_QUAL[QUAL_PARA_FC];
: 4272 5805 2          END;
: 4273 5806 2
: 4274 5807 2
: 4275 5808 2      OUTPUT_FLAGS[OUTPUT_V_FIRST] = TRUE;
: 4276 5809 2      OUTPUT_NAM = OUTPUT_FAB[FC_NAM];
: 4277 5810 2
: 4278 5811 2      OUTPUT_FLAGS[OUTPUT_NONSEQ_IN] = FALSE ;
: 4279 5812 2
: 4280 5813 2      ! If this is a compare or verify operation, get a compare buffer.
: 4281 5814 2      !
: 4282 5815 2      IF .QUAL[QUAL_COMP] THEN COMPARE_BCB = GET_BUFFER();
: 4283 5816 2
: 4284 5817 2
: 4285 5818 2      IF .QUAL[QUAL_PHYS] OR .QUAL[QUAL_IMAG]
: 4286 5819 2      THEN
: 4287 5820 3          BEGIN
: 4288 5821 3              LOCAL
: 4289 5822 3                  DESC:          VECTOR[2],          ! Descriptor for DEVCHAR
: 4290 5823 3                  DEVCHAR:      BBLOCK[DIB$K_LENGTH], ! Device characteristics
: 4291 5824 3                  STATUS;          ! Status variable
: 4292 5825 3
: 4293 5826 3
: 4294 5827 4      IF NOT (.COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV])
: 4295 5828 3      THEN
: 4296 5829 4          BEGIN
: 4297 5830 4
: 4298 5831 4              ! Strip everything but the device from the expanded string.
: 4299 5832 4              !
: 4300 5833 4              OUTPUT_NAM[NAM$B_ESL] = .BBLOCK[OUTPUT_QUAL[QUAL_DEV_DESC], DSC$W_LENGTH];
: 4301 5834 4
: 4302 5835 4
: 4303 5836 4              ! Assign a channel to the output device.
: 4304 5837 4              !
: 4305 5838 4              STATUS = ASSIGN_OUTPUT_CHANNEL(OUTPUT_QUAL[QUAL_DEV_DESC], OUTPUT_CHAN, 0, 0);
: 4306 5839 4              IF NOT .STATUS
: 4307 5840 4              THEN
: 4308 5841 4                  FILE_ERROR(
: 4309 5842 4                      BACKUP$_OPENOUT + STS$K_SEVERE,
: 4310 5843 4                      .OUTPUT_FAB,
: 4311 5844 4                      .STATUS);
: 4312 5845 3          END;
: 4313 5846 3
: 4314 5847 3
: 4315 5848 3      IF .QUAL[QUAL_PHYS]
: 4316 5849 3      THEN
: 4317 5850 4          BEGIN
: 4318 5851 4
: 4319 5852 4              ! Get the device characteristics of the output device.
: 4320 5853 4              !
: 4321 5854 4              DESC[0] = DIB$K_LENGTH;
: 4322 5855 4              DESC[1] = DEVCHAR;
: 4323 5856 4              STATUS = $GETCHN(CHAN=.OUTPUT_CHAN, PRIBUF=DESC);
: 4324 5857 4              IF NOT .STATUS
```


Restore a save set
INIT_RESTORE - initialize restore operation

E 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 143
(22)

```

4325      5858 4      THEN
4326      5859 4          FILE_ERROR(
4327      5860 4              BACKUPS_GETCHN,
4328      5861 4              .OUTPUT_FAB,
4329      5862 4              .STATUS);
4330      5863 4
4331      5864 4
4332      5865 4      ! Save the geometry data.
4333      5866 4      !
4334      5867 4      OUTPUT_DEVGEOM[DGM_SECTORS] = .DEVCHAR[DIB$B_SECTORS];
4335      5868 4      OUTPUT_DEVGEOM[DGM_TRACKS] = .DEVCHAR[DIB$B_TRACKS];
4336      5869 4      OUTPUT_DEVGEOM[DGM_CYLINDERS] = .DEVCHAR[DIB$W_CYLINDERS];
4337      5870 4      OUTPUT_DEVGEOM[DGM_MAXBLOCK] = .DEVCHAR[DIB$L_MAXBLOCK];
4338      5871 4
4339      5872 4
4340      5873 4      ! Get the bad block data.
4341      5874 4      !
4342      5875 4      IF .OUTPUT_BAD EQL 0
4343      5876 4      THEN
4344      5877 4          OUTPUT_BAD = GET_BADBLOCKS(.OUTPUT_FAB, .OUTPUT_CHAN, DEVCHAR);
4345      5878 4
4346      5879 4
4347      5880 4      ! Compute the blocks to be expected.
4348      5881 4      !
4349      5882 4      OUTPUT_FLAGS[OUTPUT_OPEN] = TRUE;
4350      5883 4      OUTPUT_BLOCK = 0;
4351      5884 4      OUTPUT_MAXBLOCK = .DEVCHAR[DIB$L_MAXBLOCK] - 1;
4352      5885 4      IF NOT .BBLOCK [DEVCHAR[DIB$L_DEVCHAR], DEV$V_RCT]
4353      5886 4      AND .DEVCHAR[DIB$L_MAXBLOCK] GTRU SMALL_DISK
4354      5887 4      THEN
4355      5888 4          OUTPUT_MAXBLOCK = .OUTPUT_MAXBLOCK -
4356      5889 4              .DEVCHAR[DIB$B_SECTORS] /
4357      5890 6              ((.DEVCHAR[DIB$B_SECTORS] *
4358      5891 6                  .DEVCHAR[DIB$B_TRACKS] *
4359      5892 5                  .DEVCHAR[DIB$W_CYLINDERS])) /
4360      5893 4                  .DEVCHAR[DIB$L_MAXBLOCK]);
4361      5894 3      END;
4362      5895 2      END;
4363      5896 1      END;

```

```

.EXTRN  SYS$GETCHN

```

39	OC	FF16	55	00000000G	00	003C	00000	9E	00002	.ENTRY	INIT RESTORE, Save R2,R3,R4,R5	:	5745
			54	00000000'	EF	9E	00009	9E	00009	MOVAB	FILE_ERROR, R5	:	
			5E	84	AE	9E	00010	9E	00010	MOVAB	OUTPUT_FAB, R4	:	
			A4	FEA0	C4	D0	00014	D0	00014	MOVAB	-124(SP), SP	:	
			C4		03	E1	0001A	E1	0001A	MOVL	QUAL+4, OUTPUT_QUAL	:	5776
			64	00BC	C4	D0	00020	D0	00020	BBC	#3, COM_FLAGS, -2\$	:	5779
				FEAB	C4	D0	00025	D0	00025	MOVL	VERIFY_FAB, OUTPUT_FAB	:	5786
					C4	95	00025	95	00025	TSTB	QUAL+15	:	5787
					09	18	00029	18	00029	BGEQ	1\$	:	
					7E	D4	0002B	D4	0002B	CLRL	-(SP)	:	5789
	D943	CF			01	FB	0002D	FB	0002D	CALLS	#1, ASSIGN_VERIFY_CHANNEL	:	
					2D	11	00032	11	00032	BRB	3\$	:	
			50	OC	A4	D0	00034	D0	00034	MOVL	OUTPUT_QUAL, R0	:	5795

RESTORE
V04-000

Restore a save set
INIT_RESTORE - initialize restore operation

F 15

16-Sep-1984 00:21:13

5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742

[BACKUP.SRC]RESTORE.B32;1

Page 144

(22)

		51	04	A0	D0	00038	MOVL	4(R0), FAB		
		50		64	D0	0003C	MOVL	OUTPUT_FAB, R0	5796	
	34	A0	34	A1	90	0003F	MOVB	52(FAB), 52(R0)		
	2C	A0	2C	A1	D0	00044	MOVL	44(FAB), 44(R0)	5797	
	009F	C0	009F	C1	90	00049	MOVB	159(FAB), 159(R0)	5798	
	00A0	C0	00A0	C1	D0	00050	MOVL	160(FAB), 160(R0)	5799	
				08	11	00057	BRB	3\$	5779	
		50	0C	A4	D0	00059	2\$: MOVL	OUTPUT_QUAL, R0	5804	
		64	04	A0	D0	0005D	MOVL	4(R0), OUTPUT_FAB		
04	A4	FC	A4	02	88	00061	3\$: BISB2	#2, OUTPUT_FLAGS	5808	
		FD	A4	8F	C1	00065	ADDL3	#148, OUTPUT_FAB, OUTPUT_NAM	5809	
				01	8A	0006E	BICB2	#1, OUTPUT_FLAGS+1	5811	
			FEA4	C4	95	00072	TSTB	QUAL+8	5815	
				0C	18	00076	BGEQ	4\$		
	00000000G	00		00	FB	00078	CALLS	#0, GET_BUFFER		
	00C8	C4		50	D0	0007F	MOVL	R0, COMPARE_BCB		
07	FEA8	C4		05	E0	00084	4\$: BBS	#5, QUAL+12, 5\$	5818	
01	FEA6	C4		03	E0	0008A	BBS	#3, QUAL+10, 5\$		
					04	00090	RET			
06	FF16	C4		03	E1	00091	5\$: BBC	#3, COM_FLAGS, 6\$	5827	
			FEA8	C4	95	00097	TSTB	QUAL+15		
				2F	19	00098	BLSS	7\$		
		51	04	A4	D0	0009D	6\$: MOVL	OUTPUT_NAM, R1	5833	
		50	0C	A4	D0	000A1	MOVL	OUTPUT_QUAL, R0		
	0B	A1	10	A0	90	000A5	MOVB	16(R0), 11(R1)		
				7E	7C	000AA	CLRQ	-(SP)	5838	
			F8	A4	9F	000AC	PUSHAB	OUTPUT_CHAN		
			10	A0	9F	000AF	PUSHAB	16(R0)		
	00000000G	00		04	FB	000B2	CALLS	#4, ASSIGN_OUTPUT_CHANNEL		
		52		50	D0	000B9	MOVL	R0, STATUS		
		0D		52	E8	000BC	BLBS	STATUS, 7\$	5839	
				52	DD	000BF	PUSHL	STATUS	5844	
				64	DD	000C1	PUSHL	OUTPUT_FAB	5843	
			00000000G	8F	DD	000C3	PUSHL	#BACKUP\$_OPENOUT+4	5842	
		65		03	FB	000C9	CALLS	#3, FILE_ERROR		
01	FEA8	C4		05	E0	000CC	7\$: BBS	#5, QUAL+12, 8\$	5848	
					04	000D2	RET			
	74	AE	74	8F	9A	000D3	8\$: MOVZBL	#116, DESC	5854	
	78	AE		6E	9E	000D8	MOVAB	DEVCHAR, DESC+4	5855	
				7E	7C	000DC	CLRQ	-(SP)	5856	
			7C	AE	9F	000DE	PUSHAB	DESC		
				7E	D4	000E1	CLRL	-(SP)		
			F8	A4	DD	000E3	PUSHL	OUTPUT_CHAN		
	00000000G	00		05	FB	000E6	CALLS	#5, SYS\$GETCHN		
		52		50	D0	000ED	MOVL	R0, STATUS		
		0D		52	E8	000F0	BLBS	STATUS, 9\$	5857	
				52	DD	000F3	PUSHL	STATUS	5862	
				64	DD	000F5	PUSHL	OUTPUT_FAB	5861	
			00000000G	8F	DD	000F7	PUSHL	#BACKUP\$_GETCHN	5859	
		65		03	FB	000FD	CALLS	#3, FILE_ERROR		
	1C	A4	08	AE	D0	00100	9\$: MOVL	DEVCHAR+8, OUTPUT_DEVGEOM	5867	
	20	A4	70	AE	D0	00105	MOVL	DEVCHAR+112, OUTPUT_DEVGEOM+4	5870	
			10	A4	D5	0010A	TSTL	OUTPUT_BAD	5875	
				12	12	0010D	BNEQ	10\$		
				5E	DD	0010F	PUSHL	SP	5877	
			F8	A4	DD	00111	PUSHL	OUTPUT_CHAN		
				64	DD	00114	PUSHL	OUTPUT_FAB		

RESTORE
V04-000

Restore a save set
INIT_RESTORE - initialize restore operation

G 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 145
(22)

00000000G	00	03	FB	00116	CALLS	#3, GET BADBLOCKS	:
10	A4	50	D0	0011D	MOVL	R0, OUTPUT_BAD	:
FC	A4	01	88	00121	BISB2	#1, OUTPUT_FLAGS	5882
18	A4	14	A4	D4 00125	CLRL	OUTPUT_BLOCK	5883
70	AE	01	C3	00128	SUBL3	#1, DEVCHAR+112, OUTPUT_MAXBLOCK	5884
2C	01	AE	E8	0012E	BLBS	DEVCHAR+1, 11\$	5885
00001000	8F	70	AE	D1 00132	CMPL	DEVCHAR+112, #4096	5886
		22	1B	0013A	BLEQU	11\$	:
	50	08	AE	9A 0013C	MOVZBL	DEVCHAR+8, R0	5891
	51	09	AE	9A 00140	MOVZBL	DEVCHAR+9, R1	:
	50		51	C4 00144	MULL2	R1, R0	:
	52	0A	AE	3C 00147	MOVZWL	DEVCHAR+10, R2	5892
	50		52	C4 0014B	MULL2	R2, R0	:
	50	70	AE	C6 0014E	DIVL2	DEVCHAR+112, R0	5893
	53	08	AE	9A 00152	MOVZBL	DEVCHAR+8, R3	5890
50	53		50	C7 00156	DIVL3	R0, R3, R0	:
	18		50	C2 0015A	SUBL2	R0, OUTPUT_MAXBLOCK	5889
			04	0015E	RET		5896

; Routine Size: 351 bytes, Routine Base: CODE + 2892

RESTORE
V04-000

Restore a save set

RESTORE_HANDLER - condition handler for restore

H 15

16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 146
(23)

```

4365 5897 1 %SBTTL 'RESTORE_HANDLER - condition handler for restore'
4366 5898 1 GLOBAL ROUTINE RESTORE_HANDLER(SIG,MECH)=
4367 5899 1
4368 5900 1 !++
4369 5901 1
4370 5902 1 FUNCTIONAL DESCRIPTION:
4371 5903 1 This routine is established as a handler for routines RESTORE, LIST,
4372 5904 1 and ANALYZE. It determines the recovery from a fatal signal.
4373 5905 1
4374 5906 1 INPUT PARAMETERS:
4375 5907 1 SIG - Standard VMS condition handler parameters.
4376 5908 1 MECH -
4377 5909 1
4378 5910 1 IMPLICIT INPUTS:
4379 5911 1 NONE
4380 5912 1
4381 5913 1 OUTPUT PARAMETERS:
4382 5914 1 NONE
4383 5915 1
4384 5916 1 IMPLICIT OUTPUTS:
4385 5917 1 NONE
4386 5918 1
4387 5919 1 ROUTINE VALUE:
4388 5920 1 VMS status code (SS$_CONTINUE or SS$_RESIGNAL).
4389 5921 1
4390 5922 1 SIDE EFFECTS:
4391 5923 1 NONE
4392 5924 1
4393 5925 1 --
4394 5926 1
4395 5927 2 BEGIN
4396 5928 2 MAP
4397 5929 2 SIG: REF BBLOCK, ! Signal parameters
4398 5930 2 MECH: REF BBLOCK; ! Mechanism parameters
4399 5931 2 EXTERNAL LITERAL
4400 5932 2 BACKUP$_SSCHANGE,
4401 5933 2 BACKUP$_READERRS,
4402 5934 2 BACKUP$_POSITERR;
4403 5935 2
4404 5936 2
4405 5937 2 IF RESTARTABLE_CONDITION(.SIG[CHFS$_SIG_NAME])
4406 5938 2 THEN
4407 5939 2 IF
4408 5940 2 .SIG[CHFS$_SIG_NAME] EQL BACKUP$_SSCHANGE OR
4409 5941 2 .SIG[CHFS$_SIG_NAME] EQL BACKUP$_READERRS OR
4410 5942 2 .SIG[CHFS$_SIG_NAME] EQL BACKUP$_FATALERR OR
4411 5943 2 .SIG[CHFS$_SIG_NAME] EQL BACKUP$_POSITERR
4412 5944 2 THEN
4413 5945 3 BEGIN
4414 5946 3 IF .RWSV_VOL_NUMBER EQL 1
4415 5947 3 THEN
4416 5948 3 GET_RESTART(.SIG, %B'110')
4417 5949 3 ELSE
4418 5950 3 GET_RESTART(.SIG, %B'111');
4419 5951 3 RETURN SS$_CONTINUE;
4420 5952 2 END ;
4421 5953 2
```

RESTORE
V04-000

Restore a save set
RESTORE_HANDLER - condition handler for restore

16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 147
(23)

: 4422
: 4423
: 4424

5954 2
5955 2 SSS_RESIGNAL
5956 1 END;

				000C 00000	.EXTRN BACKUP\$_SSCHANGE	
					.EXTRN BACKUP\$_READERRS	
					.EXTRN BACKUP\$_POSITERR	
					.ENTRY RESTORE_HANDLER, Save R2,R3	: 5898
	52	04	AC	D0 00002	MOVL SIG, R2	: 5937
	53	04	A2	D0 00006	MOVL 4(R2), R3	
			53	DD 0000A	PUSHL R3	
00000000G	00		01	FB 0000C	CALLS #1, RESTARTABLE_CONDITION	
	40		50	E9 00013	BLBC R0, 4\$	
00000000G	8F		53	D1 00016	CMPL R3, #BACKUP\$_SSCHANGE	: 5940
			1B	13 0001D	BEQL 1\$	
00000000G	8F		53	D1 0001F	CMPL R3, #BACKUP\$_READERRS	: 5941
			12	13 00026	BEQL 1\$	
00000000G	8F		53	D1 00028	CMPL R3, #BACKUP\$_FATALERR	: 5942
			09	13 0002F	BEQL 1\$	
00000000G	8F		53	D1 00031	CMPL R3, #BACKUP\$_POSITERR	: 5943
			1C	12 00038	BNEQ 4\$	
	01	00000000'	EF	B1 0003A 1\$:	CMPL RWSV_VOL_NUMBER, #1	: 5946
			04	12 00041	BNEQ 2\$	
			06	DD 00043	PUSHL #6	: 5948
			02	11 00045	BRB 3\$	
			07	DD 00047 2\$:	PUSHL #7	: 5950
			52	DD 00049 3\$:	PUSHL R2	
00000000G	00		02	FB 0004B	CALLS #2, GET_RESTART	
	50		01	D0 00052	MOVL #1, R0	: 5951
				04 00055	RET	
	50	0918	BF	3C 00056 4\$:	MOVZWL #2328, R0	: 5956
			04	0005B	RET	

; Routine Size: 92 bytes, Routine Base: CODE + 29F1

RESTORE
V04-000

Restore a save set
RESTORE - main restore routine

J 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 148
(24)

```

4426 5957 1 %SBTTL 'RESTORE - main restore routine'
4427 5958 1 GLOBAL ROUTINE RESTORE: NOVALUE=
4428 5959 1
4429 5960 1 |++
4430 5961 1
4431 5962 1 | FUNCTIONAL DESCRIPTION:
4432 5963 1 | This routine is the driver for restore, compare, and verify operations.
4433 5964 1
4434 5965 1 | INPUT PARAMETERS:
4435 5966 1 | NONE
4436 5967 1
4437 5968 1 | IMPLICIT INPUTS:
4438 5969 1 | NONE
4439 5970 1
4440 5971 1 | OUTPUT PARAMETERS:
4441 5972 1 | NONE
4442 5973 1
4443 5974 1 | IMPLICIT OUTPUTS:
4444 5975 1 | NONE
4445 5976 1
4446 5977 1 | ROUTINE VALUE:
4447 5978 1 | NONE
4448 5979 1
4449 5980 1 | SIDE EFFECTS:
4450 5981 1 | NONE
4451 5982 1
4452 5983 1 | --
4453 5984 1
4454 5985 2 BEGIN
4455 5986 2 EXTERNAL ROUTINE
4456 5987 2 INIT_IN_SAVE: NOVALUE,
4457 5988 2 READ_BUFFER,
4458 5989 2 CONCURRENT_LIST:NOVALUE,
4459 5990 2 FIN_IN_SAVE: NOVALUE;
4460 5991 2 BUILTIN
4461 5992 2 FP;
4462 5993 2
4463 5994 2 LOCAL
4464 5995 2 CHK_SAVESET : BYTE ;
4465 5996 2
4466 5997 2
4467 5998 2 | If this is not a recursive call to do a verify pass, initialize the input
4468 5999 2 | save set and establish the handler.
4469 6000 2
4470 6001 2 IF NOT .COM_FLAGS[COM_VERIFYING]
4471 6002 2 THEN
4472 6003 3 BEGIN
4473 6004 3 .FP = RESTORE_HANDLER;
4474 6005 3 INIT_IN_SAVE(FALSE);
4475 6006 3 IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .COM_FLAGS[COM_EOV] THEN RETURN;
4476 6007 3 CHK_SAVESET = 1 ;
4477 6008 3 END
4478 6009 2 ELSE CHK_SAVESET = 0 ;
4479 6010 2
4480 6011 2
4481 6012 2 | Initialize impure storage for restore.
4482 6013 2 |
```

RESTORE
V04-000

Restore a save set
RESTORE - main restore routine

K 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 149
(24)

```

4483 6014 2 INIT_RESTORE();
4484 6015 2
4485 6016 2
4486 6017 2 ! Loop over all buffers in the save set to do the operation.
4487 6018 2
4488 6019 2 WHILE (OUTPUT_BCB = READ_BUFFER()) NEQ 0 DO
4489 6020 3 BEGIN
4490 6021 3   IF .QUAL[QUAL_SS_ENCRYPT]
4491 6022 3   THEN
4492 6023 4     BEGIN
4493 6024 4       ! If we are decrypting an encrypted saveset and we don't yet have
4494 6025 4       ! the required decrypt context, do a special scan of the first block
4495 6026 4       ! to locate the backup summary record, extract the datakey information
4496 6027 4       ! and initialize the decryption context.
4497 6028 4
4498 6029 4       IF .CRYPT_DATA_CODE EQLU 0
4499 6030 5       THEN IF NOT (QUAL[QUAL_SS_ENCRYPT] = CRYPTO_INIDEC(.OUTPUT_BCB))
4500 6031 4       THEN SIGNAL(BACKUP$_BACNOTENC);
4501 6032 4     END
4502 6033 3   ELSE
4503 6034 4     BEGIN
4504 6035 4
4505 6036 4       ! Make sure that the save set is not encrypted.
4506 6037 4
4507 6038 4       IF .CHK_SAVESET
4508 6039 4       THEN IF CRYPTO_CHKSAV(.OUTPUT_BCB)
4509 6040 4       THEN SIGNAL(BACKUP$_ENCSAVSET);
4510 6041 4       CHK_SAVESET = 0;
4511 6042 3     END;
4512 6043 3
4513 6044 3   ! If this is an encrypted saveset, decrypt the buffer
4514 6045 3
4515 6046 3   IF .QUAL[QUAL_SS_ENCRYPT]
4516 6047 3   THEN CRYPTO_DECR_BLOCK(.OUTPUT_BCB);
4517 6048 3
4518 6049 3   IF .QUAL[QUAL_LIST] AND NOT .COM_FLAGS[COM_VERIFYING]
4519 6050 3   THEN CONCURRENT_LIST(.OUTPUT_BCB);
4520 6051 3   DEBLOCK(.OUTPUT_BCB, RESTORE_ONE_RECORD);
4521 6052 3   OUTPUT_BCB = 0;
4522 6053 2   END;
4523 6054 2
4524 6055 2
4525 6056 2 ! Finish the operation.
4526 6057 2
4527 6058 2 IF .COM_FLAGS[COM_VERIFYING] THEN OUTPUT_FLAGS[OUTPUT_V_LAST] = TRUE;
4528 6059 2 FIN RESTORE();
4529 6060 2 OUTPUT_FLAGS[OUTPUT_V_LAST] = FALSE;
4530 6061 2
4531 6062 2
4532 6063 2 ! If this is not a verify pass, do the termination for the input save set.
4533 6064 2 ! Also, note that nothing was restored if this is the case.
4534 6065 2
4535 6066 2 IF NOT .COM_FLAGS[COM_VERIFYING]
4536 6067 2 THEN
4537 6068 3   BEGIN
4538 6069 3     FIN_IN_SAVE(FALSE);
4539 6070 3   IF NOT .QUAL[QUAL_PHYS]
```

RESTORE
V04-000

Restore a save set
RESTORE - main restore routine

L 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 150
(24)

: 4540
: 4541
: 4542
: 4543
: 4544
: 4545
: 4546

```
6071 3      AND NOT .COM_FLAGS[COM_FILESEEN]
6072 3      AND NOT .INPUT_FLAGS[INPUT_WILDSAVE]
6073 3      THEN
6074 3          SIGNAL(
6075 3              BACKUP$_NOFILES, 1, BBLOCK[.QUAL[QUAL_INPU_LIST], QUAL_EXP_DESC]);
6076 2      END;
6077 1  END;
```

.EXTRN INIT_IN_SAVE, READ_BUFFER
.EXTRN CONCURRENT_LIST
.EXTRN FIN_IN_SAVE

```
.ENTRY RESTORE, Save R2,R3,R4
MOVAB LIB$SIGNAL, R4
MOVAB OUTPUT_BCB, R3
BBS #3, COM_FLAGS, 2$
MOVAB RESTORE_HANDLER, (FP)
CLRL -(SP)
CALLS #1, INIT_IN_SAVE
BBC #6, INPUT_FLAGS, 1$
BLBC COM_FLAGS, 1$
RET
MOVAB #1, CHK_SAVESET
BRB 3$
CLRB CHK_SAVESET
CALLS #0, INIT_RESTORE
CALLS #0, READ_BUFFER
MOVL R0, OUTPUT_BCB
BEQL 10$
BBC #4, QUAL+14, 5$
TSTL CRYPT_DATA_CODE
BNEQ 7$
PUSHL OUTPUT_BCB
CALLS #1, CRYPTO_INIDEC
INSV R0, #4, #1, QUAL+14
BLBS R0, 7$
PUSHL #BACKUP$_BACNOTENC
CALLS #1, LIB$SIGNAL
BRB 7$
BLBC CHK_SAVESET, 6$
PUSHL OUTPUT_BCB
CALLS #1, CRYPTO_CHKSAV
BLBC R0, 6$
PUSHL #BACKUP$_ENCSAVSET
CALLS #1, LIB$SIGNAL
CLRB CHK_SAVESET
BBC #4, QUAL+14, 8$
PUSHL OUTPUT_BCB
CALLS #1, CRYPTO_DECR_BLOCK
BLBC QUAL+11, 9$
BBS #3, COM_FLAGS, 9$
PUSHL OUTPUT_BCB
CALLS #1, CONCURRENT_LIST
PUSHAB RESTORE_ONE_RECORD
PUSHL OUTPUT_BCB
```

```
54 00000000G 00 001C 00000
53 00000000' EF 9E 00002
1E FFOE C3 03 E0 00010
6D 8B AF 9E 00016
7E D4 0001A
06 00000000G 00 01 FB 0001C
FF5C C3 06 E1 00023
01 FFOE C3 E9 00029
52 04 0002E
01 90 0002F 1$:
02 11 00032
52 94 00034 2$:
FE0A CF 00 FB 00036 3$:
00000000G 00 00 FB 0003B 4$:
63 50 D0 00042
79 13 00045
24 FEA2 C3 04 E1 00047
057C C3 D5 0004D
38 12 00051
63 DD 00053
01 00000000G 00 01 FB 00055
04 50 F0 0005C
25 50 E8 00063
00000000G 8F DD 00066
64 01 FB 0006C
1A 11 0006F
15 52 E9 00071 5$:
63 DD 00074
00000000G 00 01 FB 00076
09 09 50 E9 0007D
00000000G 8F DD 00080
64 01 FB 00086
52 94 00089 6$:
09 FEA2 C3 04 E1 0008B 7$:
63 DD 00091
00000000G 00 01 FB 00093
0F FE9F C3 E9 0009A 8$:
09 FFOE C3 03 E0 0009F
63 DD 000A5
00000000G 00 01 FB 000A7
F1BA CF 9F 000AE 9$:
63 DD 000B2
```

```
: 5958
:
: 6001
: 6004
: 6005
:
: 6006
:
: 6007
: 6001
: 6009
: 6014
: 6019
:
: 6021
: 6029
:
: 6030
:
: 6031
: 6021
: 6038
: 6039
:
: 6040
:
: 6041
: 6046
: 6047
:
: 6049
: 6050
: 6051
```


RESTORE
V04-000

Restore a save set
RESTORE - main restore routine

M 15
16-Sep-1984 00:21:13
5-Sep-1984 21:46:26

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]RESTORE.B32;1

Page 151
(24)

00000000G	00	02	FB	000B4	CALLS	#2, DEBLOCK	:	
		63	D4	000BB	CLRL	OUTPUT_BCB	:	6052
		FF7B	31	000BD	BRW	4\$	:	6019
04	FF0E	C3	03	E1	000C0	10\$: BBC	:	6058
	F4	A3	04	88	000C6	BISB2	:	
	E01A	CF	00	FB	000CA	11\$: CALLS	:	6059
	F4	A3	04	8A	000CF	BICB2	:	6060
2C	FF0E	C3	03	E0	000D3	BBS	:	6066
			7E	D4	000D9	CLRL	:	6069
00000000G	00	01	FB	000DB	CALLS	#1, FIN IN SAVE	:	
1D	FEA0	C3	05	E0	000E2	BBS	:	6070
17	FF0E	C3	02	E0	000E8	BBS	:	6071
11	FF5C	C3	06	E0	000EE	BBS	:	6072
7E	FE94	C3	08	C1	000F4	ADDL3	:	6075
			01	DD	000FA	PUSHL	:	
			8F	DD	000FC	PUSHL	:	
			03	FB	00102	CALLS	:	
			04	00105	12\$: RET		:	6077

; Routine Size: 262 bytes, Routine Base: CODE + 2A4D

RESTORE Restore a save set
V04-000 RESTORE - main restore routine
: 4548 6078 1 END
: 4549 6079 0 ELUDOM

N 15
16-Sep-1984 00:21:13 VAX-11 Bliss-32 V4.0-742
5-Sep-1984 21:46:26 [BACKUP.SRC]RESTORE.B32;1

Page 152
(25)

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
DATA	172	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	11091	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	198	1	1000	00:02.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RESTORE/OBJ=OBJ\$:RESTORE MSRC\$:RESTORE/UPDATE=(ENH\$:RESTORE)

Size: 10907 code + 2480 data bytes
Run Time: 03:02.3
Elapsed Time: 09:25.5
Lines/CPU Min: 2000
Lexemes/CPU-Min: 24265
Memory Used: 846 pages
Compilation complete

0013 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

